NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

SEARCH PARAMETER OPTIMIZATION FOR DISCRETE, BAYESIAN, AND CONTINUOUS SEARCH ALGORITHMS

by

Benjamin Kalkwarf

September 2017

Thesis Advisor:	Hong Zhou
Second Reader:	Bard Mansager

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704–0188

1. AGENCY USE ONLY *(Leave Blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	September 2017	Master's Thesis 09-14-2015 to 09-22-2017

4. TITLE AND SUBTITLE

SEARCH PARAMETER OPTIMIZATION FOR DISCRETE, BAYESIAN, AND CONTINUOUS SEARCH ALGORITHMS

5. FUNDING NUMBERS

6. AUTHOR(S)

Benjamin Kalkwarf

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Postgraduate School
Monterey, CA 93943

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

N/A

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

The views expressed in this document are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol Number: N/A.

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release. Distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

Search and Detection Theory is the overarching field of study that covers many scenarios. These range from simple search and rescue acts to prosecuting aerial/surface/submersible targets on mission. This research looks at varying the known discrete and Bayesian algorithm parameters to analyze the optimization. It also expands on previous research of two searchers with search radii coupled to their speed, executing three search patterns: inline spiral search, inline ladder search, and a multipath ladder search. Analysis reveals that the Bayesian search and discrete search work similarly, but the Bayesian search algorithm provides a more useful output in location probability. Results from the continuous search were similar to previous research, but variance in time to detection became more complex than basic increasing or decreasing ranges.

14. SUBJECT TERMS

Search Theory, Undersea Warfare, South China Sea, Anti-Submarine Warfare

15. NUMBER OF PAGES 253

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2–89)
Prescribed by ANSI Std. 239–18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

SEARCH PARAMETER OPTIMIZATION FOR DISCRETE, BAYESIAN, AND CONTINUOUS SEARCH ALGORITHMS

Benjamin Kalkwarf
Lieutenant, United States Navy
B.S., Iowa State University, 2007

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN APPLIED MATHEMATICS

from the

NAVAL POSTGRADUATE SCHOOL
September 2017

Approved by: Hong Zhou
Thesis Advisor

Bard Mansager
Second Reader

Craig Rasmussen
Chair, Department of Applied Mathematics

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Search and Detection Theory is the overarching field of study that covers many scenarios. These range from simple search and rescue acts to prosecuting aerial/surface/submersible targets on mission. This research looks at varying the known discrete and Bayesian algorithm parameters to analyze the optimization. It also expands on previous research of two searchers with search radii coupled to their speed, executing three search patterns: inline spiral search, inline ladder search, and a multipath ladder search. Analysis reveals that the Bayesian search and discrete search work similarly, but the Bayesian search algorithm provides a more useful output in location probability. Results from the continuous search were similar to previous research, but variance in time to detection became more complex than basic increasing or decreasing ranges.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

List of Figures

List of Tables

List of Acronyms and Abbreviations

AAW	Anti-Air Warfare
ASW	Anti-Submarine Warfare
ASWORG	Anti-Submarine Warfare (ASW) Operations Research Group
ASUW	Anti-Surface Warfare
CASP	computer-aided search program
CNA	Center for Naval Analyses
ISR	Intelligence, Surveillance and Reconnaissance
JW	Joint Warfare
MIW	Mine Warfare
NPS	Naval Postgraduate School
SDT	Search and Detection Theory
SOF	Special Operation Forces
SW	Strike Warfare
USCG	United States Coast Guard
USN	U.S. Navy
USV	unmanned surface vehicle
UUV	unmanned underwater vehicle
WWII	World War II

THIS PAGE INTENTIONALLY LEFT BLANK

Acknowledgments

First, I would like to take a moment to thank the U.S. Navy and the Submarine Force for allowing me the opportunity to attend NPS and pursue my master's degree.

I thank Dr. Hong Zhou for serving as my thesis advisor, Bard Mansager for being my second reader, and Susan Hawthorne for being my thesis processor. I would also like to thank Professor John Joseph, Dr. Daphne Kapolka, and RADM Winford G. Ellis for their expertise in undersea warfare.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1:
Introduction

The process of finding the location of one or more unknown subjects is known as Search and Detection Theory (SDT) and is of great interest to U.S. Navy (USN). While there are many applications, both military and non-military, SDT is of particular import to Anti-Submarine Warfare (ASW). This is due to the desire to locate adversarial submarines because the nature of submarines is to use their covert operating patterns to conduct Intelligence, Surveillance and Reconnaissance (ISR), Anti-Surface Warfare (ASUW), Strike Warfare (SW) and Special Operation Forces (SOF) missions.

With the advent of high end diesel-electric submarines, increased tensions and activity from capable nuclear submarines (Russia and China), and the proliferation of submarine technologies to previously non-capable navies, the need for better approaches to SDT is evident. The proliferation of these submarine technologies, and the rise of autonomous vehicles such as unmanned underwater vehicles (UUV) and unmanned surface vehicles (USV), both as preprogrammed searchers and potential targets, dictate that the parameters of SDT algorithms must be investigated.

1.1 History

The Operations Evaluation Group (now known as the Center for Naval Analyses (CNA)) [1] was created during World War II (WWII) to enlist the help of mathematicians and scientists to assist in developing new ASW tactics and strategies. This small group of MIT scientists worked with ASW Operations Research Group (ASWORG) directly in the field for optimum observations and real-time solutions [2]. These scientists used the burgeoning search theory to improve U-boat search and neutralization, especially with favorable patterns and operating characteristics for patrol aircraft.

After the war, search theory was extended to other naval operations such as searching for the USS *Scorpion* (SSN-589), lost ordnance, and tracking Russian ballistic missile submarines [3]. The United States Coast Guard (USCG) developed a computer-aided search program (CASP) [4] for missing boats and mariners lost at sea.

Dr. Bernard Koopman of the WWII ASWORG developed the historical basis for search theory by dictating the following basic principles.

1. Quantify the probability of the sensor capability to detect the target.
2. Classify the movement of the searchers and target.
3. Analyze the constraints on the searcher's forces and resources
4. Develop the highest probability of detection with the greatest economy of forces [5].

1.2 Applications and Previous Research

The applications for SDT are numerous, so that while the context of this research is limited to an ASW submersible search problem, the basics can be extended to various problem sets and models. Research conducted at Naval Postgraduate School (NPS) has focused on naval, non-naval, and other Joint Warfare (JW) applications. Research has also been conducted at non-military institutions for civilian applications.

1.2.1 Some Previous Research

1. Zachary Lukens's 2016 thesis [6] investigated SDT applied to a single target in the South China Sea and two searchers using cookie cutter sensors utilizing inline spiral, inline ladder, and multipath ladder searches. His research found that increasing either the detection radii or speeds decreased the time to detection and that spiral search was faster than ladder searches at large radii, but both ladder searches were faster than spiral search for small radii.
2. Volkan Sozen's 2014 thesis [7] used SDT applied to Unmanned Aerial Vehicles (UAV) patrolling a border. He modeled the search as one UAV searching the area, two UAVs searching the same area, and two UAVs independently searching two equally split sub-areas. His research found that detection increased with the second searcher and splitting the searchers into independent sub-areas, the border was covered more efficiently.
3. Dr. Michael Atkinson [8] (together with Moshe Kress and Rutger-Jan Lange) researched searching for a target discrete location based on receiving of intelligence of varying reliability and the time cost incurred based on searching a certain area. His research became the problem of looking at the probability of detection based on

amount of intelligence gathered weighed against the target escaping due to a delayed search. This has direct applications to military and non-military problems based on the information quality. The study concluded that high intelligence credibility had better results in waiting to attack, while low credibility had better results for immediate attack.

4. H. Zhou and H. Wang in [9] examined optimum paths for a solitary searcher looking for a solitary target using spiral, ladder, loop and square spiral paths.They showed that a stationary searcher is optimally placed in the center of the search area. It also showed that, with a moving searcher, the ladder pattern gives the most likely detection. In [10], they examined optimum searcher placement seeking a target travelling back and forth between a detectable operating area and an undetectable hiding area. They showed closed form solutions for determining mean time to detection based on the placement of the searcher in a given path from hiding to operating area or within the operating area itself. Their results showed how to distribute sensors ("searchers") among the routes and the operating area. They extended their study to include sensors in detecting a target moving between a hiding area and an operating area [11]. Based on the average time to detection, they were able to evaluate the performance of placing the sensor(s) to monitor various travel pathway(s) or to scan the operating area.

1.2.2 Applications

The USN has utilized SDT for

- ASW
- ASUW
- Anti-Air Warfare (AAW)
- SW
- Mine Warfare (MIW)
- other operations.

Other branches and JW applications include

- ISR
- Counter-drug operations
- Counter-terrorism operations

- Ballistic Missile Defense
- SW and Artillery Fires operations
- Search and Rescue.

Multiple civilian applications are also prevalent in research and development, to include:

- Search and Rescue
- Quality Control
- Resource exploration and search (oil, minerals, fishing, etc.)
- Microeconomics (resource to market matching).

CHAPTER 2:
Discrete Search

2.1 Theory

The following sections outline the basic concepts and theory behind the discrete search theory.

2.1.1 Terms

A - Total search area (no searches conducted outside of A)

n - Total number of equal area subdivisions (boxes) of A : ($i \in \{1..n\}$)

$\boldsymbol{p_i}$ - P[tgt in ith box] probability of target present in ith box

$\boldsymbol{q_i}$ - P[miss detection|tgt in ith box] probability of failing to detect target in ith box given it is present in ith box

$\boldsymbol{k_i}$ - number of *glimpses* (discrete time interval) with search was conducted in ith box

m - total number of glimpses: $m = \sum_{i=1}^{n} k_i$

By the concept of total probability, $\sum_{i=1}^{n} p_i = 1$.

2.1.2 General Concepts

The basic assumptions for the discrete search model are [12]:

- Single stationary target
- Single searcher with imperfect sensor (q_i)
- Total search time limited (m glimpses)
- No time lost transitioning from one box to another
- Each search conducted independent of other glimpses
- No false positives
- Non-adaptive search.

The probability of the target not being detected in the ith box on a given glimpse is

$P[\text{miss detection}] = P[\text{miss detection} \mid \text{tgt in } i\text{th box}]P[\text{tgt in } i\text{th box}] = P_M^i = p_i q_i$ and the detection probability on a given glimpse for the ith box is $P[\text{detection}] = P[\text{detection} \mid \text{tgt in } i\text{th box}]P[\text{tgt in } i\text{th box}] = P_D^i = p_i(1 - q_i)$. After k_i glimpses, the probabilities of missing or detecting in the ith box are $P_M^{i,k_i} = p_i q_i^{k_i}$ and $P_D^{i,k_i} = p_i(1 - q_i^{k_i})$, respectively. This leads to [13], by the total probability concept, the likelihood of total failure and detection as

$$P_M = \sum_{i=1}^{n} p_i q_i^{k_i} \tag{2.1}$$

and

$$P_D = \sum_{i=1}^{n} p_i(1 - q_i^{k_i}). \tag{2.2}$$

2.1.3 Model Overview

The greedy algorithm is a myopic search model that is designed to attempt to optimize Equation 2.1 in order to streamline the detection based on a limited number of resources (time, searchers, etc.). For each glimpse, the probability of detection is calculated given previous glimpses have failed to detect the target. This change in detection probability [13]

$$\Delta = (1 - q_i^{k_i+1})p_i - (1 - q_i^{k_i})p_i = (1 - q_i)q_i^{k_i} p_i \tag{2.3}$$

is used to select the maximum detection probability and the search is conducted in that box, increasing k_i by 1 for that box. This updates the probability for that box alone. If the target is not found, the search proceeds to the next glimpse using the same formula. This will be repeated until the target is found or $\sum_{i=1}^{n} k_i = m$. The cumulative detection probability is the sum of the maximum Δ for at every step or Equation 2.3.

2.1.4 Example

The example in Table 2.1 from [12], outlines the "greedy" search algorithm. The target is in one of three boxes, with associated location and conditional miss probabilities. The probability of detection for each cell is calculated based on finding the target on the first glimpse with no previous looks ($\vec{k} = (0, 0, 0)$) so that $q_i^{k_i} = 1$. The highest probability is in the first cell making the $q_i^{k_i} < 1$ reducing the probability on the next round of glimpses.

The cumulative probability after six glimpses is the sum of the maximum probabilities at each round, $P_D = .25 + .125 + .1 + .08 + .064 + .0625 = .6815$.

Table 2.1. Optimal Sequence of Discrete Looks. Adapted from [12].

p_1	p_2	p_3
.5	.4	.1
q_1	q_2	q_3
.5	.8	0

(k_1, k_2, k_3)	$p_1(1-q_1)q_1^{k_1}$	$p_2(1-q_2)q_2^{k_2}$	$p_3(1-q_3)q_3^{k_3}$
(0,0,0)	**.25**	.08	.1
(1,0,0)	**.125**	.08	.1
(2,0,0)	.0625	.08	**.1**
(2,0,1)	.0625	**.08**	0
(2,1,1)	.0625	**.064**	0
(2,2,1)	**.0625**	.0512	0

2.2 Problem and Model

This basic model for discrete search is based solely on the conditional miss probabilities (q_i) and the location probabilities (p_i). These parameters will be varied and results shown.

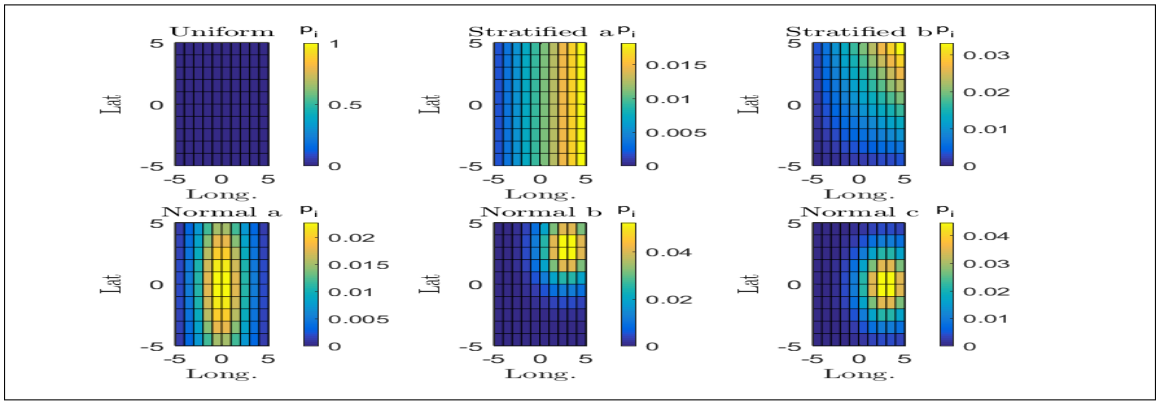
2.2.1 Parameter

The location probabilities will be varied according to Figure 2.1 and the detection probabilities will be varied according to Figure 2.2. The uniform location and conditional detection probabilities are to set up the base case to compare other distribution combinations too.

The base location probability per box, $p_0 = \frac{1}{n} = \frac{1}{100}$ as shown in the uniform case. Two linear stratified location probability densities are used to simulate the target being placed preferentially along some line of longitude (small scale) for the first distribution or some line of constant longitude/latitude line. Each of these increase by a factor of 10 over the search area in the direction of greatest change. Three Gaussian distributions are used to simulate the likelihood of the target having a well-localized location probability prior. They follow a distribution function $p(x, y) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-x_0)^2}{2\sigma_x^2} - \frac{(y-y_0)^2}{2\sigma_y^2}}$. The centered Gaussian has different σ_{lat} and σ_{long}. The other two Gaussian plots are centered on a corner and side with $\sigma_{lat} = \sigma_{long}$.

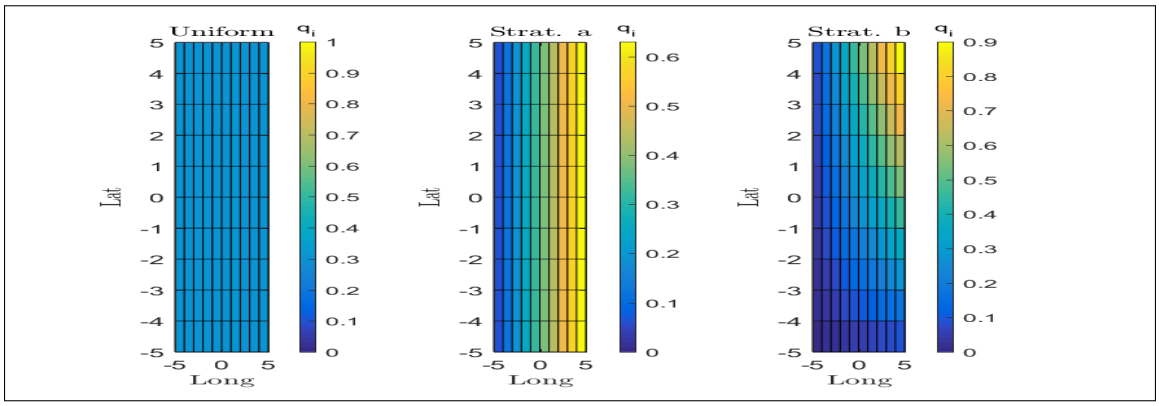
The conditional detection probability densities are uniform as a base sensor that works the same in all locations $q_i = .3$, giving a 70% of detection at each location. The two stratified detection probability densities are linear stratified along latitude or longitude from $q_i = (.063, .63)$. This simulates the sensor performance based on some geographic profile (i.e., change in bottom type for SONAR).

Figure 2.1. Location Probability Distributions $\boldsymbol{p_i}$



From top left clockwise: Uniform dist., Linear stratified along longitude, Linear stratified along Long/Lat, Gaussian dist., Gaussian centered in corner, Gaussian centered on side

Figure 2.2. Conditional Miss Probability Distributions $\boldsymbol{q_i}$



From left to right: Uniform dist., Linear stratified along longitude, Linear stratified along latitude

2.3 Results

The following sections show the results for six different location probability densities and the three conditional search probability densities. Each combination was examined for $n = 50$ glimpses (half the number of available boxes).

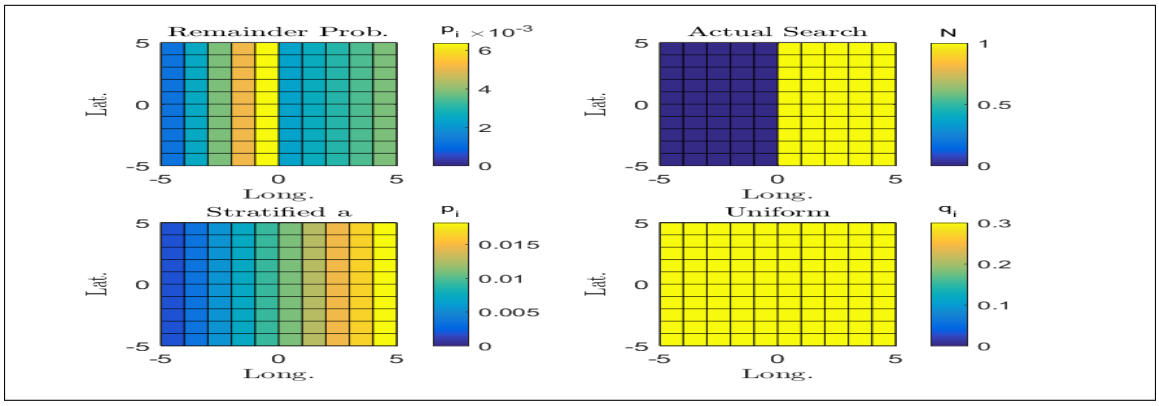
Since there are 18 possible combinations each with various tables and figures, the following will show a small sample of the results. All of the figures and tables will be presented in Appendices A and B, respectively.

2.3.1 Linear Stratified Location

Figures 2.3, 2.4, and 2.5 show the remaining location probability density for the three conditional search probability densities after 50 glimpses. Also shown are the boxes that were selected to search by the greedy algorithm.

For the uniform conditional search probability in Figure 2.3, unsurprisingly, the selected boxes start with the highest p_i and slowly work to the left as p_i lowers.

Figure 2.3. Stratified Location and Uniform Search Density Results

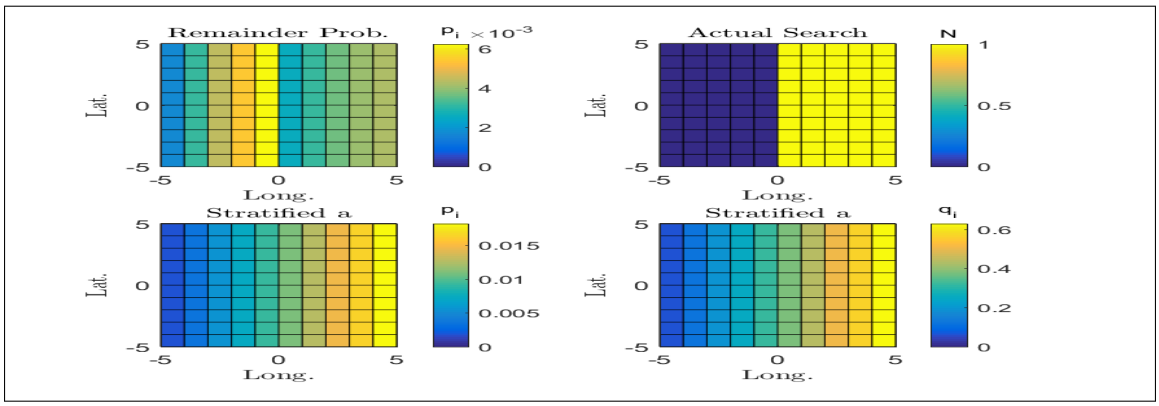


Remaining probability of detection and boxes search based on stratified p_i and uniform q_i

For the longitudinal stratified conditional search probability in Figure 2.4, the selected boxes also started with highest p_i and slowly work to the left as p_i lowers. This is slightly unexpected since a better probability of detection exists in areas of lower target location probability. But the factor of $1 - q_i$ is 93.7% in the western side and 37% in the east. This

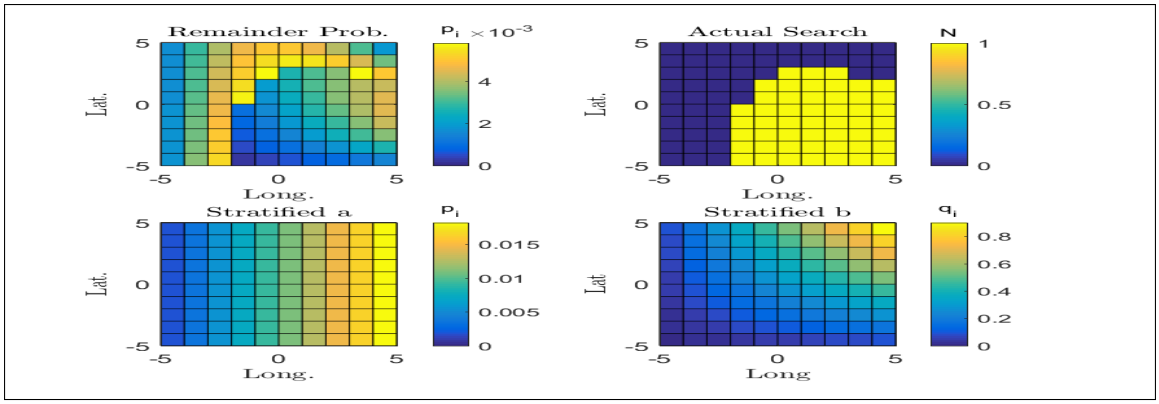
coupled with the p_i being a factor of 10 higher in the east leads it to be a higher probability of detection even though the sensor performs worse there.

Figure 2.4. Stratified Location and Longitudinal Stratified Search Results



Remaining probability of detection and boxes search based on stratified p_i and linear stratified q_i

Figure 2.5. Stratified Location and Corner Stratified Search Density Results



Remaining probability of detection and boxes search based on stratified p_i and corner stratified q_i

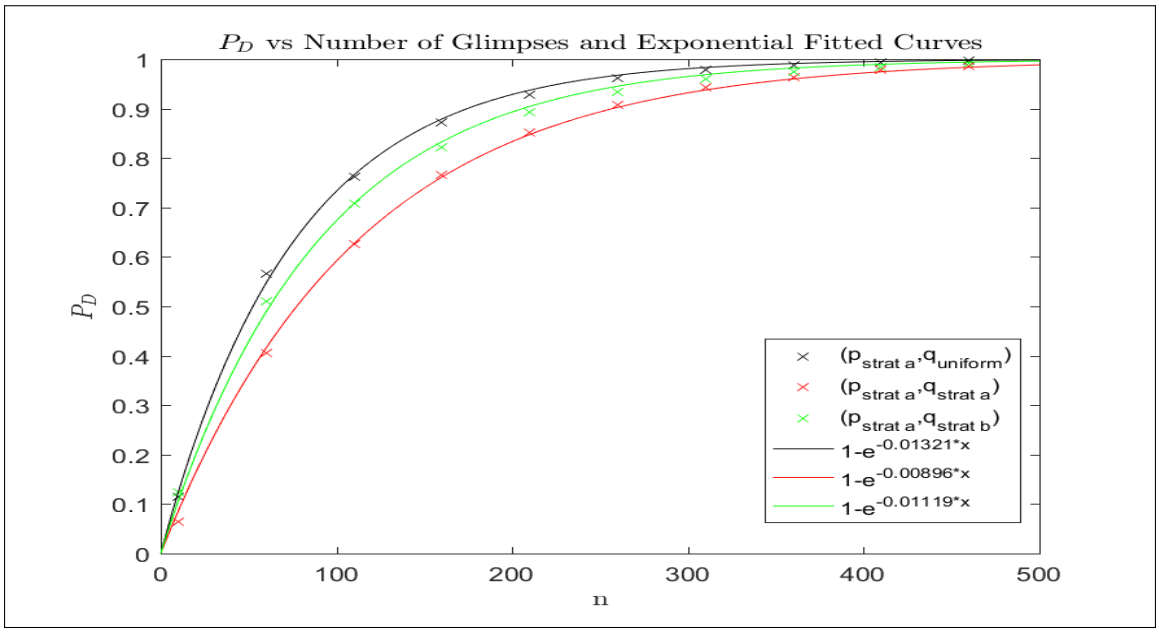
For the corner stratified conditional search probability in Figure 2.5, the selected boxes start in the southeast due to the juxtaposition of high p_i and low q_i. The path moves mostly along discretized radial arcs centered on the southeast corner. Which box is chosen via the greedy algorithm is dependent on when $1 - q_i$ will push a probability of detection higher for a lower p_i.

For the longitudinal stratified p_i location probability density, the following results were found based on running the search for up to 500 glimpses. A data set using a step size of 5 is shown in Figure 2.6 along with associated fit lines as calculated from MATLAB's fit algorithm based on:

$$f(x) = 1 - e^{-bx}.$$

Each of the fit lines are color coded to the data set with black for uniform, red for longitudinal stratification, and green for corner stratification. As the number of glimpses increases, the probability approaches 1, with the uniform case increasing faster than the other two.

Figure 2.6. Stratified Location Cumulative Probability and Fit Lines



Cumulative probabilities of detection based on the number of glimpses. Included are the fit best fit line

Table 2.2. Parameter Variation for p_i for Longitudinal Stratification

Parameter	b	r^2	SSE	RMSE
p_{strat},$q_{uniform}$	0.013208	0.99908	0.00065853	0.008554
p_{strat},q_{long}	0.0089597	0.99920	0.00066179	0.0085751
p_{strat},q_{corner}	0.011187	0.99808	0.0013515	0.012254

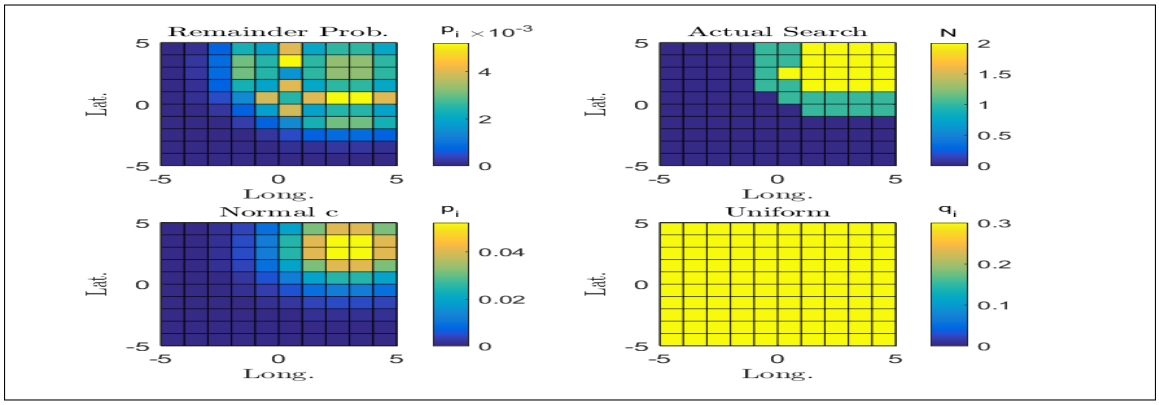
Best fit line parameters for a longitudinally stratified location probability density of the form $f(x) = 1 - e^{-bx}$

2.3.2 Gaussian Corner Location

The remaining location probability densities for the three conditional search probability densities after 50 glimpses are shown in Figures 2.7, 2.8, and 2.9. Also shown are the boxes that were selected to search by the greedy algorithm.

For the uniform conditional search probability in Figure 2.7, unsurprisingly, the selected boxes start with highest p_i and slowly work outward as p_i depletes. A second look begins occurring in those higher probability boxes prior to searching all boxes.

Figure 2.7. Gaussian Corner Location and Uniform Search Density Results

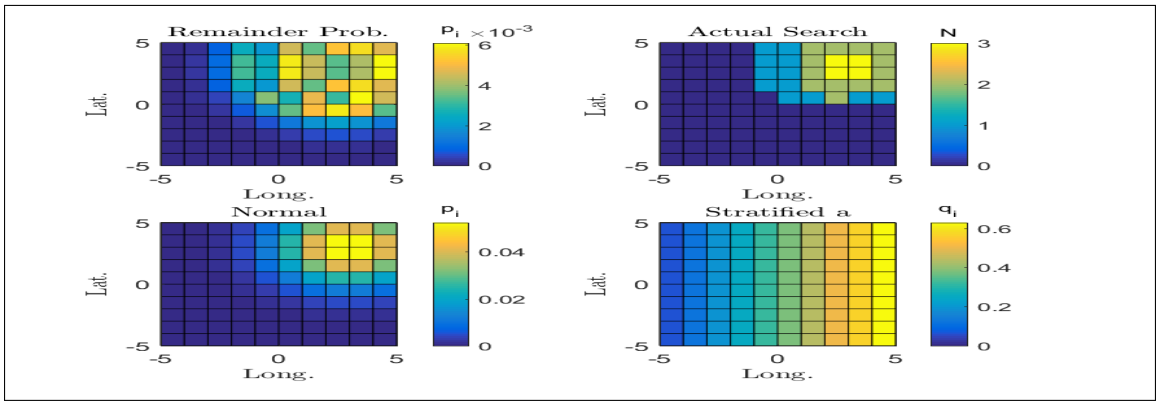


Remaining probability of detection and boxes search based on Gaussian p_i and uniform q_i

For the longitudinal stratified conditional search probability in Figure 2.8, the selected boxes also started with highest p_i and slowly works outward as p_i lowers. This is slightly

unexpected since there is better probability of detection in some areas of lower target location probability. As the probability of detection decreases on subsequent looks, the lower q_i increases the likelihood of finding the target to the west of the initial location locus. From the Remainder Probability graph in Figure 2.8, the higher location probability remains dispersed in the northwest corner albeit non-uniformly.

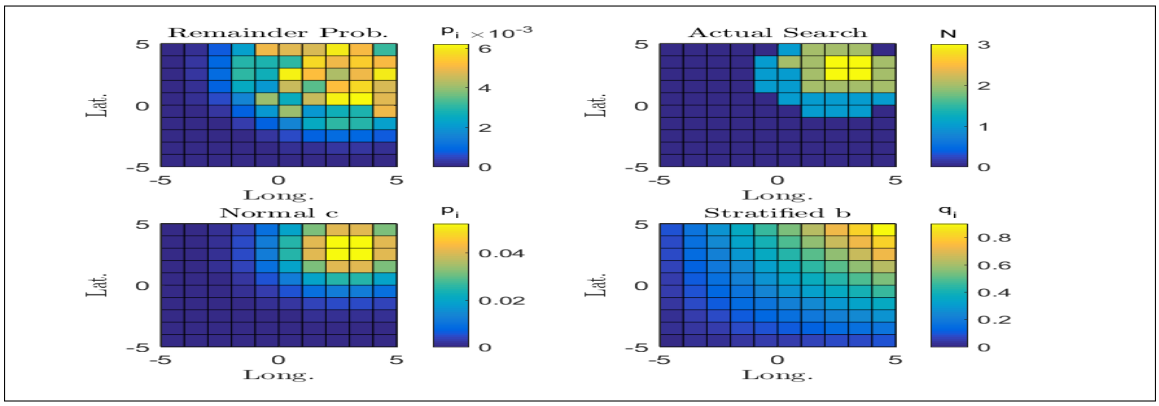
Figure 2.8. Gaussian Corner and Longitudinal Stratified Search Density Results



Remaining probability of detection and boxes search based on stratified p_i and linear stratified q_i

For the corner stratified conditional search probability in Figure 2.9, the selected boxes start in the northeast due to the high p_i overriding the high q_i. The Remainder Probability graph in Figure 2.9 shows some areas 2-3 times greater than other areas, especially around the original locus. Some areas will be searched multiple times before every box is searched once.

Figure 2.9. Gaussian Corner and Corner Stratified Search Density Results



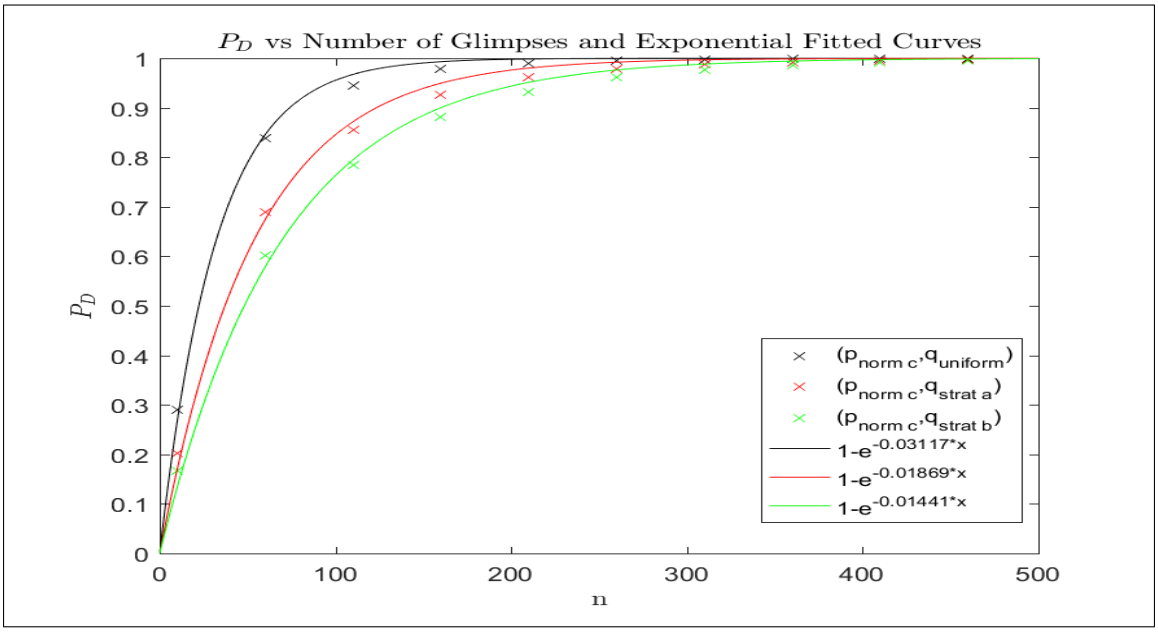
Remaining probability of detection and boxes search based on stratified p_i and corner stratified q_i

For the Gaussian Corner p_i location probability density, the following results were found based on running the search for up to 500 glimpses. A data set using a step size of 5 is plotted in Figure 2.10 along with associated fit lines as calculated from MATLAB's fit algorithm based on:

$$f(x) = 1 - e^{-bx}.$$

Each of the fit lines are color coded to the data set with black for uniform, red for longitudinal stratification, and green for corner stratification. As the number of glimpses increase, the probability approaches 1, with the uniform case increasing faster than the other two.

Figure 2.10. Gaussian Corner Cumulative Probability and Fit Lines



Cumulative probabilities of detection based on the number of glimpses. Included are the fit best fit lines

Table 2.3. Parameter Variation for p_i for Corner Gaussian Location

Parameter	b	r^2	SSE	RMSE
p_{GC},$q_{uniform}$	0.03117	0.99683	0.0013941	0.012446
p_{GC},q_{long}	0.01869	0.9952	0.0027014	0.017325
p_{GC},q_{corner}	0.01441	0.99523	0.0029724	0.018173

Best fit line parameters for a Gaussian Corner location probability density of the form $f(x) = 1 - e^{-bx}$

2.3.3 Cumulative Detection Probability

The total probability of detection for the after 50 glimpses is shown in Table 2.4. The Gaussian p_i's have higher probabilities, which is not unexpected since they have a much higher concentration on a few points. This allows for multiple searches over a smaller number of points instead of 50 different boxes. The other result seems to be that higher

probability results when p_i matches up to $1 - q_i$ as it does when $p_{stratified}$ has a higher probability of detection when paired with q_{corner_strat} instead of q_{long_strat}.

Table 2.4. P_D after 50 glimpses

p_i \ q_i	Uniform	Long. Stratified	Corner Stratified
Uniform	0.35	0.4055	0.44906
Long. Stratified	0.50909	0.34927	0.46119
Corner Stratified	0.57157	0.43958	0.40031
Gaussian	0.55932	0.52965	0.5919
Gaussian Side	0.71881	0.55738	0.65176
Gaussian Corner	0.79975	0.63655	0.54999

Total probability of detection after 50 glimpses based on various combinations

CHAPTER 3:
Bayesian Search

3.1 Theory

The following sections outline the basic concepts and theory behind the Bayesian approach discrete search theory.

3.1.1 Terms

The same terms from Chapter 2 apply along with the following random variables:

$\boldsymbol{L_i}$ - Target is located in box i

$\boldsymbol{M_i}$ - Target is not detected in box i

$\boldsymbol{p_0}$ - P[tgt not in A] probability of target not in search area

By the concept of total probability, $\sum_{i=1}^{n} p_i = P_A$ and $1 - P_A = p_0$.

3.1.2 General Concepts

We use the same basic assumptions of Section 2.1.2 with an added assumption that the target may not be in the search area. The basics of Bayesian search theory is to update the probability of the target being located in the ith (p_i) box based on previous failures to detect. Searching the ith box and failing to find it will decrease its likelihood of being located there and increase the probability of its being located in another box. This posterior probability comes from Bayes' Theorem.

$$P(L_i \mid M_i) = \frac{P(M_i \mid L_i)P(L_i)}{P(M_i)} \tag{3.1}$$

The probability of not detecting the target in the ith box ($P(M_i)$) is currently the only uncalculated quantity. It is, by total probability, the sum of the conditional detection failures based on the location probabilities. $P(M_i) = P(M_i \mid L_i)P(L_i) + P(M_i \mid \neg L_i)P(\neg L_i)$. $P(L_i)$ and $\neg P(L_i)$ are p_i and $1 - p_i$, respectively. $P(M_i \mid L_i) = q_i$ and $P(M_i \mid \neg L_i) = 1$ based

on the assumption of now false positives. This updates Equation 3.1 to

$$p_i' = P(L_i \mid M_i) = \frac{q_i p_i}{(1 - p_i) + p_i q_i}. \tag{3.2}$$

The probability that the target is located in jth box ($j \neq i$) must also be updated since it is more likely that it is not located in the ith box,

$$P(L_j \mid M_i) = \frac{P(M_i \mid L_j)P(L_j)}{P(M_i)}, \tag{3.3}$$

where $P(L_i) = p_j$ and $P(M_i) = (1 - p_i) + p_i q_i$ as found above. $P(M_i \mid L_j)$ is the conditional probability of failing to detect the target in box i given it is in box j. Since there are no false positives, the searcher will always fail to detect when searching in a box where the target is not located, making $P(M_i \mid L_j) = 1$. This updates the location probability by

$$p_{j\neq i}' = P(L_j \mid M_i) = \frac{p_j}{(1 - p_i) + p_i q_i}. \tag{3.4}$$

This updates all $p_{i\neq j}$ including p_0.

3.1.3 Model Overview

In the Bayesian search model, the conditional probabilities of detection failure only factor into updating the new location probabilities. As such, the search algorithm searches the box with the highest likelihood of containing the target and searches. Upon failure to find, it updates all location probabilities and repeats until the target is found or the number of glimpses has been reached.

This still follows the Bayesian search algorithm from Chapter 2 in that at each step the probability to detect $P_{Di} = p_i(1 - q_i)$ is calculated for all boxes. The search is conducted in the box with the highest detection probability. After a failure to detect, the p_i's are updated via Equation 3.2 and Equation 3.4 [14]. The Bayesian search algorithm then treats the next glimpse as the first with the updated location probabilities [14].

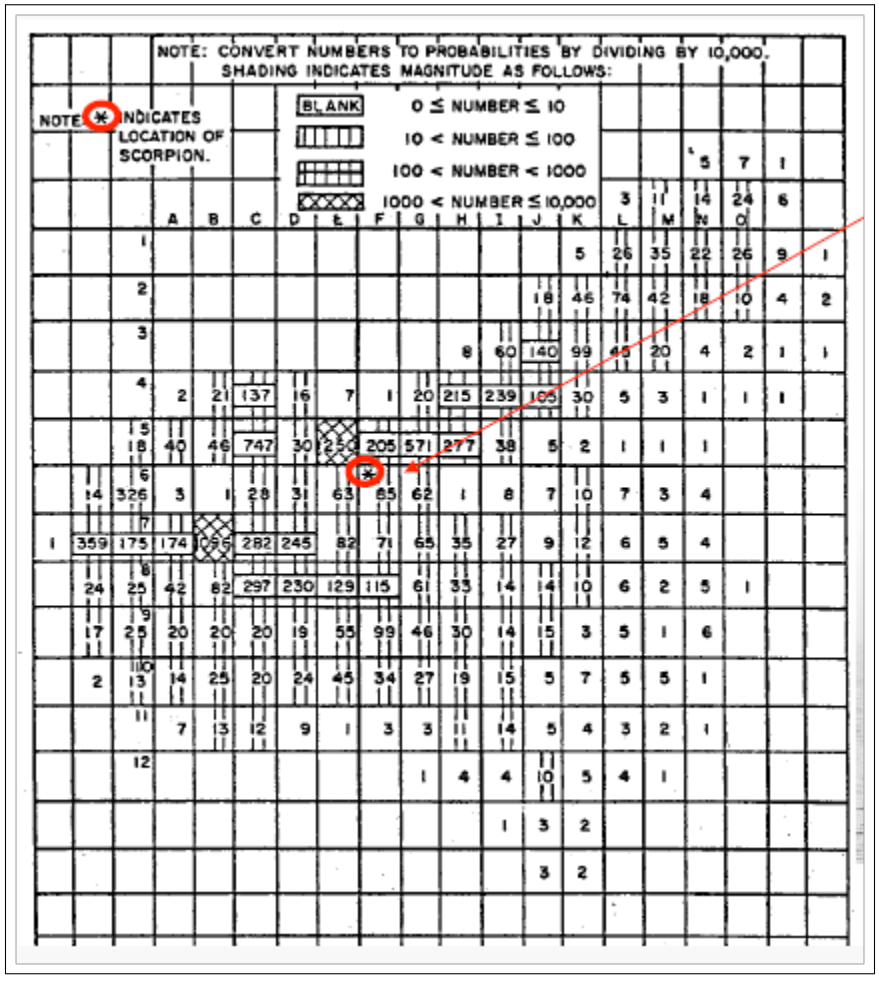
If $p_0 = 0$, this will provide the same output as the Discrete Search Method. For example, the updated location probabilities are just Δ_i from the Discrete Method divided by $1 - q_i$'s and then normalized so that the sum is equal to 1. Multiplying $1 - q_i$'s back in after

normalization will make the same evaluation as in Table 2.1 [14].

3.1.4 Examples from Other Research

The Bayesian method has been used for several real-world searches. Famously used in USS *Scorpion*, a submarine lost in 1968, it was also used in MV *Derbyshire*, the SS *Central America*, the lost hydrogen bomb from a B-52 crash, Air France flight 447, and most recently flight MH370. Figure 3.1 is the location probability density originally constructed for the USS *Scorpion* [15].

Figure 3.1. *USS Scorpion* Location Probability. Source: [15].



The basic steps outlined in previous examples are:

1. Construct plausible conjectures as to the cause of the loss of the object.
2. For each conjecture, construct a location probability density map ($P(L_i) = p_i$).
3. Build the conditional detection probability maps for every location covered under the location probability map ($P(M_i \mid L_i) = q_i$). This will be a function of sensor's ability to detect in the given environment and must be built individually based on current conditions and sensor dynamics.
4. Combine the maps in Step 2 and 3 to create an overall detection probability density map ($p_i(1 - q_i)$).
5. Develop a search path that scans the highest to lowest probability areas.
6. Revise the location probabilities continuously during the search. If multiple conjectures are being tested simultaneously, each should be updated independently and if the search fails to corroborate aspects of a particular conjecture the likelihood of that scenario will need to be downgraded [15].

One last aspect of the Bayesian search, that is not being explored in this research, is the concept of the cost of a search. If, for example, searching one particular box takes longer time, requires more resources, or is more dangerous, then that should be taken into account as a weight on the search. If the cost function can be estimated as c_i, then the new detection probabilities come out as $\frac{p_i(1-q_i)}{c_i}$ [16].

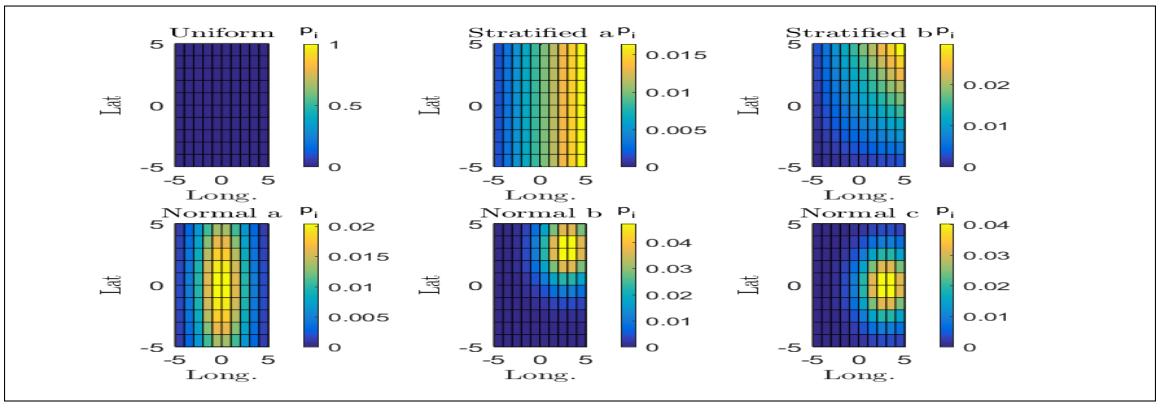
3.2 Problem and Model

This basic model for Bayesian search is based solely on the conditional miss probabilities (q_i) and the initial location probabilities (p_{oi}). These parameters will be varied and results shown.

3.2.1 Parameters

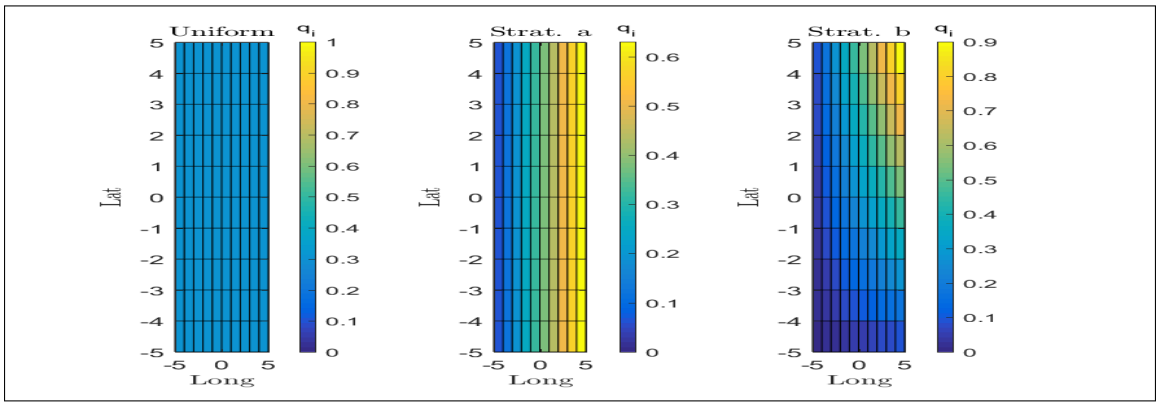
The location and conditional miss will have the same variations as presented in Chapter 2 (Figures 2.1 and 2.2) and they are repeated in Figures 3.2 and 3.3 for the reader's convenience with $p_0 = .1$ and the detection probabilities will be varied according to Figure 2.2, which are shown again here for clarity.

Figure 3.2. Bayesian Initial Location Probabilities p_{oi}



From top left clockwise: Uniform dist., Linear stratified along longitude, Linear stratified along Long/Lat, Gaussian dist., Gaussian centered in corner, Gaussian centered on side

Figure 3.3. Bayesian Miss Probabilities q_i



From left to right: Uniform dist., Linear stratified along longitude, Linear stratified along latitude

3.3 Results

The following sections show the results for six different location probability densities and the three conditional search probability densities. Each combination was examined for $n = 50$ glimpses (half the number of available boxes).

The tables and figures showing all 18 possible combinations are presented in Appendix A

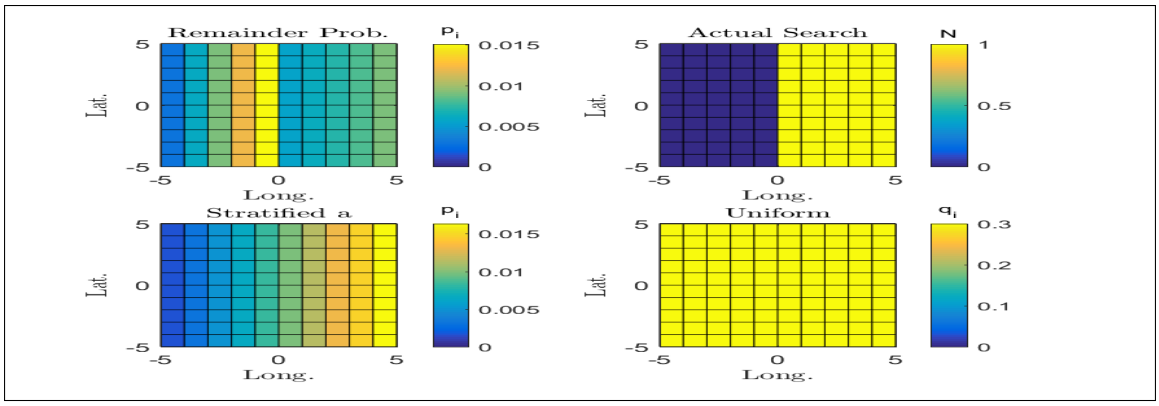
and Appendix B, respectively.

3.3.1 Linear Stratified Location

Figures 3.4, 3.5, and 3.6 show the remaining location probability densities for the three conditional search probability densities after 50 glimpses. Also shown are the boxes that were selected to search by the Bayesian search algorithm.

For the uniform conditional search probability in Figure 3.4, unsurprisingly, the selected search boxes start with highest p_i and slowly work to the left as p_i lowers. This follows the same pattern as the discrete search from Chapter 2. The exception being that the upper-leftmost graph of Figure 3.4 now shows the updated location probabilities in Bayes' algorithm instead of merely the next search step as in the discrete search method. This follows as the Bayesian method and Discrete method calculate roughly the same probability of detection with the exception that the Bayesian search modifies $p_0 = .1 \rightarrow 0.1846$.

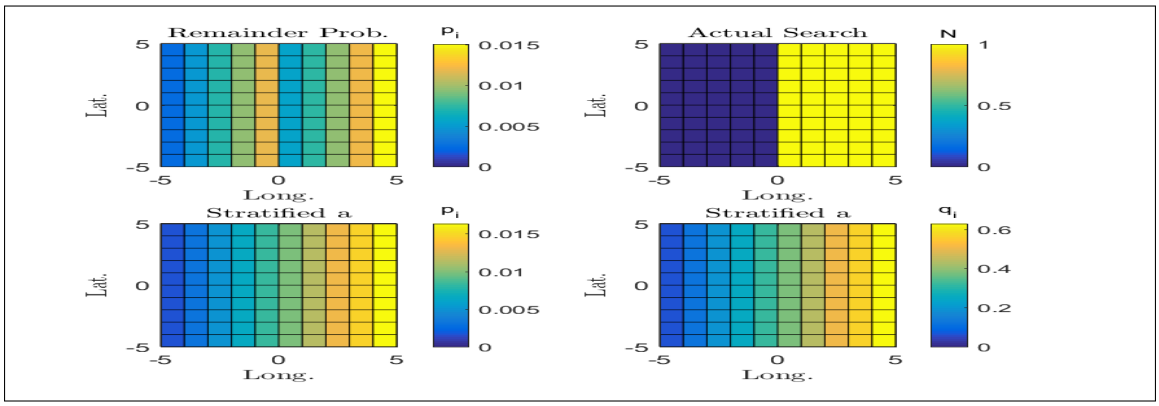
Figure 3.4. Stratified Location and Uniform Search Density Results



Remaining probability of detection and boxes search based on stratified p_i and uniform q_i

For the longitudinal stratified conditional search probability in Figure 3.5, the selected search boxes also start with highest p_i and slowly work to the left as p_i lowers. This follows the slightly unexpected results from the discrete search of Chapter 2 possibly from the factor of $1 - q_i$ being 93.7% in the western side and 37% in the east. This coupled with the p_i being a factor of 10 higher in the east than the west leads it to be a higher probability of detection even though the sensor performs worse there. The probability of the target not being in the search area $p_0 = .1 \rightarrow 0.1458$.

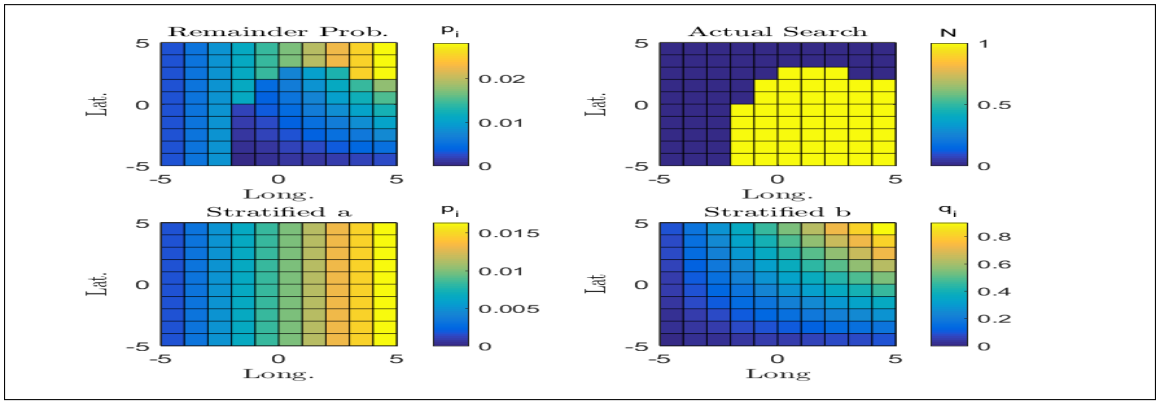
Figure 3.5. Stratified Location and Longitudinal Stratified Search Results



Remaining probability of detection and boxes search based on stratified p_i and linear stratified q_i

For the corner stratified conditional search probability in Figure 3.6, the selected boxes start in the southeast due to the juxtaposition of high p_i and low q_i. The path moves mostly along discretized radial arcs centered on the southeast corner. Which box is chosen via the Bayesian search algorithm is dependent on when $1 - q_i$ will push a probability of detection higher for a lower p_i. The probability of the target not being in the search area $p_0 = .1 \rightarrow 0.1710$.

Figure 3.6. Stratified Location and Corner Stratified Search Density Results



Remaining probability of detection and boxes search based on stratified p_i and corner stratified q_i

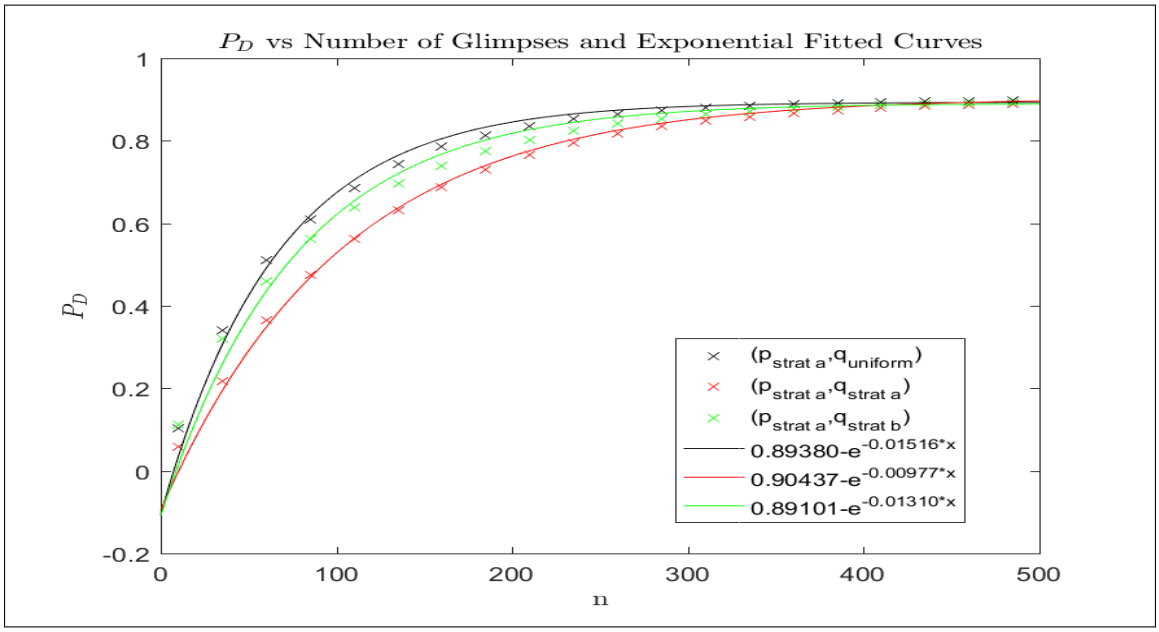
For the longitudinal stratified p_i location probability density, the following results were

found based on running the search for up to 500 glimpses or a total probability of detection at 95%. A data set using a step size of 5% is plotted in Figure 3.7 along with associated fit lines as calculated from MATLAB's fit algorithm based on

$$f(x) = a - e^{-bx}.$$

Each of the fit lines is color coded to the data set with black for uniform, red for longitudinal stratification, and green for corner stratification. As the number of glimpses increases, the probability approaches $P_A = .9$, with the uniform case increasing faster than the other two. The associated parameters for the best fit lines are presented in Table 3.1.

Figure 3.7. Stratified Location Cumulative Probability and Fit Lines



Cumulative probabilities of detection based on the number of glimpses. Included are the fit best fit lines

The results are very similar to results from Chapter 2 as the same decision algorithm is used; however, the exponential growth approaches $P_A = .9$ the initial probability that the target is in the search area A.

Table 3.1. Parameter Variation for p_i for Longitudinal Stratification

Parameter	a	b	r^2	SSE	RMSE
p_{strat},$q_{uniform}$	0.8938	-0.015155	0.99013	0.0086678	0.0093465
p_{strat},q_{long}	0.90437	-0.0097729	0.99496	0.0053481	0.017237
p_{strat},q_{corner}	0.89101	-0.0131	0.97996	0.017876	0.031514

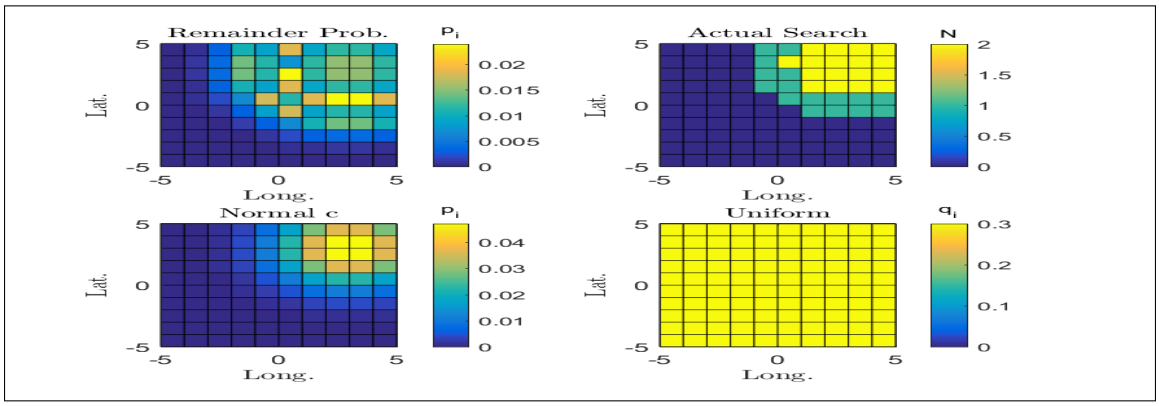
Best fit line parameters for a longitudinally stratified location probability density of the form $f(x) = a - e^{-bx}$

3.3.2 Gaussian Corner Location

Figures 3.8, 3.9, and 3.10 show the remaining location probability densities for the three conditional search probability densities after 50 glimpses. Also shown are the boxes that were selected to search by the Bayesian search algorithm.

For the uniform conditional search probability in Figure 3.8, unsurprisingly, the selected boxes started with highest p_i and slowly works outward as p_i shifts higher probability to other boxes. A second look begins occurring in those higher probability boxes prior to searching all boxes.

Figure 3.8. Gaussian Corner Location and Uniform Search Density Results

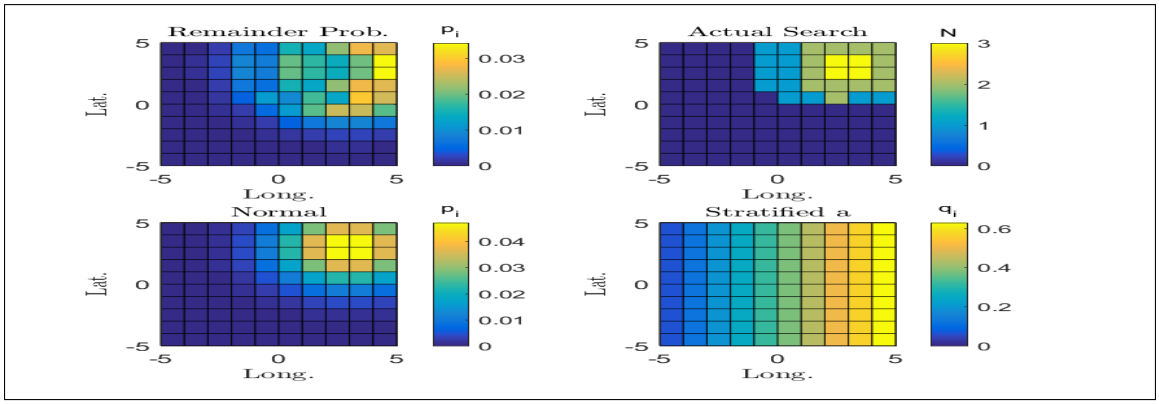


Remaining probability of detection and boxes search based on Gaussian p_i and uniform q_i

For the longitudinal stratified conditional search probability in Figure 3.9, the selected boxes also started with highest p_i and slowly work outward as p_i lowers. This follows the slightly unexpected result from Chapter 2 since there are better probabilities of detection in some

areas of lower target location probability. As the location probability shifts to other boxes, the higher p_i increases the likelihood of the target being located to the west of the initial location locus. From the Remainder Probability graph in Figure 3.9, the higher location probability remains dispersed in the northwest corner albeit non-uniformly.

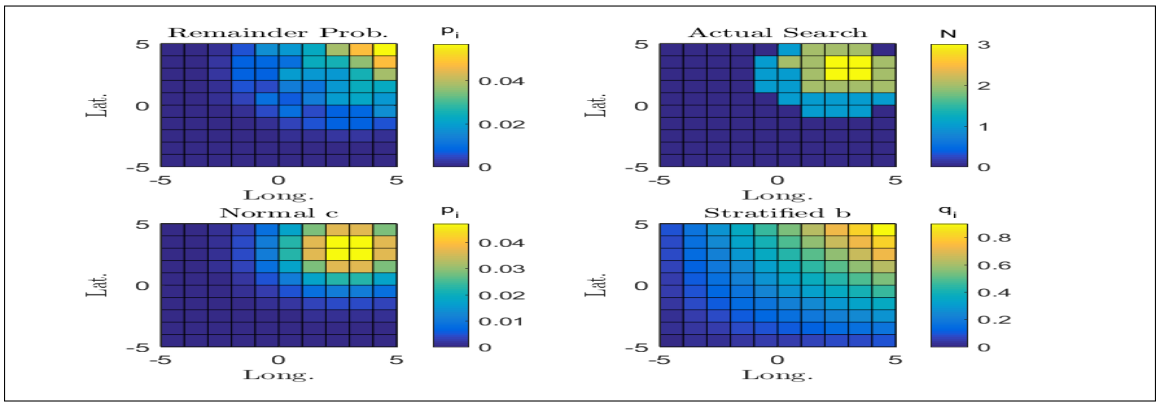
Figure 3.9. Gaussian Corner and Longitudinal Stratified Search Density Results



Remaining probability of detection and boxes search based on stratified p_i and linear stratified q_i

For the corner stratified conditional search probability in Figure 3.10, the selected boxes start in the northeast due to the high p_i overriding the high q_i. The Remainder Probability graph in Figure3.10 shows some areas 2-3 times greater than other areas, especially around the original locus. Some boxes have been searched twice or three times before every box has been searched once. Since the location probability distribution p_i remains identical to that from Chapter 2 (other than the being 90% of the original), the overall patterns of the search are identical to the results from Chapter 2.

Figure 3.10. Gaussian Corner and Corner Stratified Search Density Results



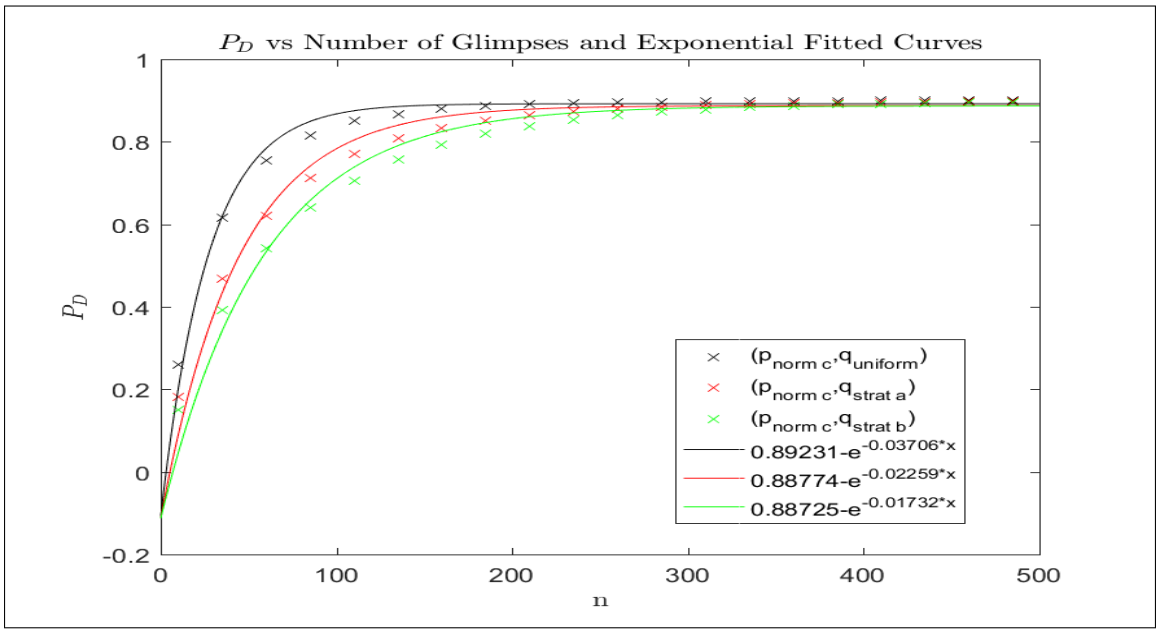
Remaining probability of detection and boxes search based on stratified p_i and corner stratified q_i

For the Gaussian corner p_i location probability density, the following results were found based on running the search for up to 500 glimpses or a total probability of detection at 95%. A data set using a step size of 5% is plotted in Figure 3.11 along with associated fit lines as calculated from MATLAB's fit algorithm based on

$$f(x) = a - e^{-bx}.$$

Each of the fit lines are color coded to the data set with black for uniform, red for longitudinal stratification, and green for corner stratification. As the number of glimpses increase, the probability approaches $P_A = .9$, with the uniform case increasing faster than the other two. The associated parameters for the best fit lines are presented in Table 3.2.

Figure 3.11. Gaussian Corner Cumulative Probability and Fit Lines



Cumulative probabilities of detection based on the number of glimpses. Included are the fit best fit lines

Table 3.2. Parameter Variation for p_i for Corner Gaussian Distribution

Parameter	a	b	r^2	SSE	RMSE
p_{GC},$q_{uniform}$	0.89231	-0.03706	0.98424	0.0068884	0.019563
p_{GC},q_{long}	0.88774	-0.022588	0.97697	0.014581	0.028462
p_{GC},q_{corner}	0.88725	-0.01732	0.9747	0.018924	0.032424

Best fit line parameters for a corner Gaussian location probability density of the form $f(x) = a - e^{-bx}$

The results are very similar to results from Chapter 2 as the same decision algorithm is used.

3.3.3 Cumulative Detection Probability

The total probability of detection produced by the Bayesian search algorithm after 50 glimpses is shown in Table 3.3. The Gaussian p_i's have higher probabilities, which is not

unexpected since they have a much higher concentration on a few points. This allows for multiple searches over a smaller number of points instead of 50 different boxes. The other result seems to be that higher probability results when p_i matches up to $1 - q_i$ as it does when $p_{stratified}$ has a higher probability of detection when paired with q_{corner_strat} instead of q_{long_strat}. This follows the same pattern as the Discrete Search Method but lower overall since there is some probability that the target is outside of A. The increase to p_0 after 50 glimpses is shown in Table 3.4.

Table 3.3. P_D after 50 glimpses

p_i \ q_i	Uniform	Long. Stratified	Corner Stratified
Uniform	0.315	0.36495	0.40415
Long. Stratified	0.45818	0.31435	0.41507
Corner Stratified	0.51441	0.39562	0.36028
Gaussian	0.50339	0.47669	0.53271
Gaussian Side	0.64693	0.50164	0.58658
Gaussian Corner	0.71978	0.57289	0.49499

Total probability of detection after 50 glimpses based on various combinations

Table 3.4. p_0 after 50 glimpses

p_i \ q_i	Uniform	Long. Stratified	Corner Stratified
Uniform	0.1460	0.1575	0.1678
Long. Stratified	0.1846	0.1458	0.1710
Corner Stratified	0.2059	0.1655	0.1563
Gaussian	0.2014	0.1911	0.2140
Gaussian Side	0.2832	0.2007	0.2419
Gaussian Corner	0.3569	0.2341	0.1980

Probability of target not in A after 50 glimpses based on various combinations

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 4:
Continuous Search

4.1 Theory

The following section outlines the basic concepts and theory behind a continuous exhaustive search.

Terms

A - Total search area (no searches conducted outside of A)

$\boldsymbol{v_S}$ - Searcher's velocity

$\boldsymbol{v_T}$ - Target's velocity

R - Range of the sensor. Cookie cutter sensor that will detect the target if the range is less than R. $R = R(v_S)$

w - Sweep width. $w = 2R$

The sensor is a "cookie cutter" sensor that has 100% chance of detecting the target if the distance from the searcher to the target is less than R. In [6] both R and both v_S were independent of each other and varied accordingly. Here, R is a function of v_S which more closely resembles real-world sensor performance, especially in SONAR sensors.

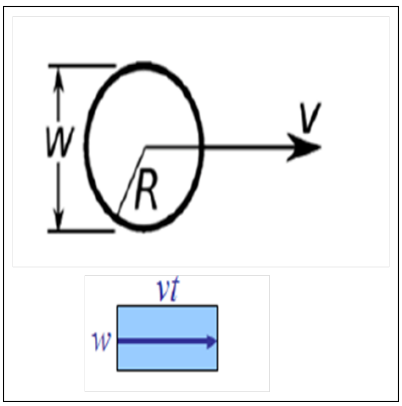
4.1.1 General Concepts

The basic assumptions for this continuous search model are [6] [9] [13]:

- Single target undergoing Brownian diffusion
- Two searchers with "cookie-cutter" sensors
- Exhaustive search using one of three search patterns
- Sensors radius is velocity dependent $R(v_S)$
- Each searcher results independent of the other
- No false positives
- Non-adaptive search.

As shown in Figure 4.1 the sweep width for a moving searcher ($w = 2R$) will cover an area searched $A(t) = wvt$. The exhaustive search covers all of A without overhanging any previously searched area or any area outside of A. The total time to coverage then is $t^* = \frac{A}{vw}$ [6].

Figure 4.1. Search Area and Sweep Width. Source: [9].



Search Area (A) as it relates to Sweep Width (w)

4.1.2 Model Overview

Based on the work done by H. Wang and H. Zhou [9] and Z. Lukens's thesis [6], the two searchers will move along one of three set paths: the inline ladder, the multipath ladder and the inline spiral search. For each path setup, 100,000 independent Monte Carlo simulations are conducted with the target uniformly distributed over A. The independent targets move randomly with a Brownian motion with a given diffusion rate.

The searchers will move along the given path independently of each other without coordination. An input parameter of the initial separation of the two searcher is given so that the searchers don't overlap throughout the search. The separation in the inline searches was estimated at $4L_x$ where L_x is half the width of A [6].

4.2 Problem and Model

The model is thus two searchers (aircraft, surface vessels, subs, etc.) of similar capability performing the inline spiral in Figure 4.2, the inline ladder search in Figure 4.3, and the

multipath ladder search in Figure 4.4.

Figure 4.2. Spiral Search. Source: [9].

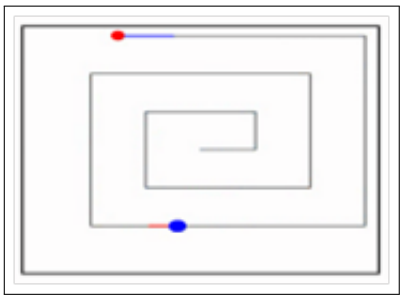
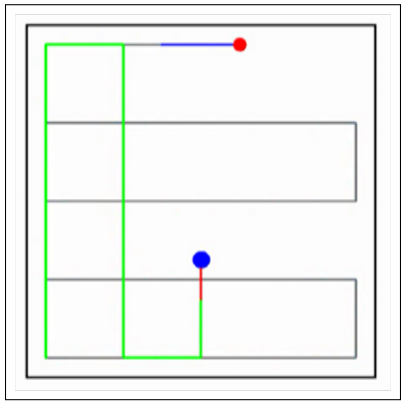


Figure 4.3. Inline Ladder Search. Source: [9].

Figure 4.4. Multipath Ladder Search Pattern. Source: [9].



4.2.1 Parameter

As an extension of previous work done [6], this chapter will explore the search patterns with a sensor that obeys Equation 4.1.

$$R(V_S) = R_0 e^{-V_S/V} \tag{4.1}$$

R_0 is the performance of the sensor at $v_S = 0$ and V is the exponential decay factor based on the sensor performance at speed. Each sensor, in reality, will behave differently based on its characteristics. For instance, a towed SONAR array requires a minimum speed in order to be drooped correctly and at the proper depth, whereas a hull-mounted SONAR does not have this minimum effect and is only limited at higher speed by self-noise. Further

compounding the real life sensors is that, at higher speeds, the self noise has non-linear effects based on cavitation above a threshold. To simplify the issue, the sensor is assumed to be operating in the range where Equation 4.1 holds. Each search velocity will be varied as $v_S \in (8, 9, 10, 11, 12)$ and the velocity decay factor from $V \in (5, 6, 7, 8, 9, 10)$ giving 150 different combinations for each search pattern. The initial search radius, $R_0 = 2.5$ nm, will be set based on the largest search radius in [6].

4.3 Results

In Tables 4.1, 4.2, and 4.3, the time in hours to achieve a probability of detection at 90% ($P_D = 90\%$) are shown for $V = 9$ the inline ladder search, the multipath ladder, and the spiral search, respectively.

Table 4.1. Time to $P_D = 90\%$ $(V = 9)$

		Ship 1				
	Velocities	8	9	10	11	12
Ship 2	8	6.8662	6.9782	7.1117	7.5250	7.7083
	9	7.5069	7.8241	7.7043	7.8639	7.9255
	10	9.7362	9.0589	8.7202	8.4741	8.3054
	11	10.1011	10.0333	9.2198	8.9486	8.6854
	12	10.0425	10.1511	10.0968	9.2826	8.9569

Inline ladder search

Table 4.2. Time to $P_D = 90\%(V = 9)$

		Ship 1				
	Velocities	8	9	10	11	12
	8	8.1866	8.2470	8.2123	8.1351	8.0883
	9	9.3043	8.9871	8.7202	8.3385	8.1969
Ship 2	10	10.3288	9.5669	9.2282	8.7452	8.4140
	11	10.7790	10.0333	9.4231	9.0164	8.6854
	12	10.4768	10.5311	9.7168	9.1197	8.7940

Multipath ladder search

Table 4.3. Time to $P_D = 90\%(V = 9)$

		Ship 1				
	Velocities	8	9	10	11	12
	8	9.6390	9.3043	9.6515	10.2367	10.3139
	9	9.8330	9.4100	9.3129	9.7621	9.8254
Ship 2	10	10.4135	9.4822	9.3129	9.5587	9.6625
	11	10.5078	10.0333	9.5587	9.6943	9.7168
	12	10.3682	11.6710	11.5625	10.6397	10.5854

Spiral search

In a marked difference from the results found in [6], faster search speed does not result in a quicker time to reach $P_D = .9$ in all cases. For instance, the lowest time to detection for the inline ladder search is with $v_1 = 8$ and $v_2 = 8$, for multipath ladder $v_1 = 12$ and $v_2 = 8$, and for spiral search $v_1 = 9$ and $v_2 = 8$. In fact, keeping $v_2 = 8$ and varying v_1 over the range causes an increase and then decrease. Similar results happen on some other variations but as can be seen it is not consistent from one speed to another. Another noteworthy result is that varying the speed of ship 1 has less of an effect normally than varying ship 2's speed. An example of this can be seen in the multipath ladder. Raising v_1 from 8 to 12 knots keeps

the time to detection relatively constant around 8 hours, while raising v_1 from 8 to 12 knots increase the time to detect to approximately 8.5 hours.

Similar to the results found in [6], the inline ladder search has the lowest time to detect, followed by the multipath ladder, and finally the spiral search for a given set of parameters. As expected, increasing V only lowers the time to detection. This follows [6] as increasing V raises the search radii without changing the speed. Direct comparison to other results from [6] is problematic. This work varies three of the four parameters from Equation 4.1, directly varying v_1 and v_2 and indirectly R_1 and R_2. In [6], R_1 and R_2 are varied directly. If Equation 4.1 is constructed with a V and R_0 so that R_1 and R_2 equal values from [6], then results will be the same as the search algorithm work in the same manner.

A similar presentation for the various searches is outlined below in Figures 4.5-4.10.

Spiral Search

Figure 4.5. P_D vs Time

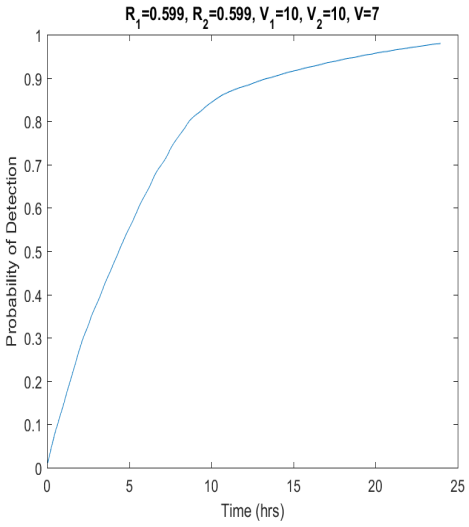
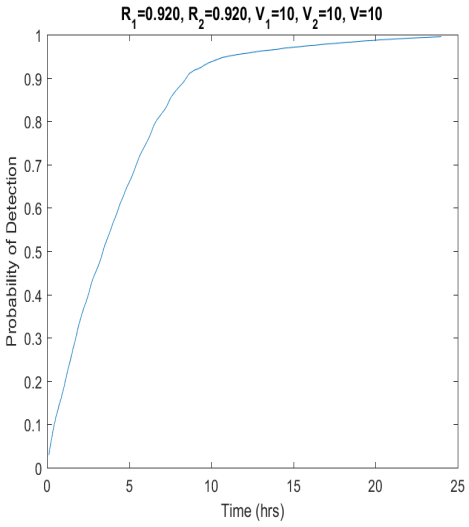


Figure 4.6. P_D vs Time



Multipath Ladder

Figure 4.7. P_D vs Time

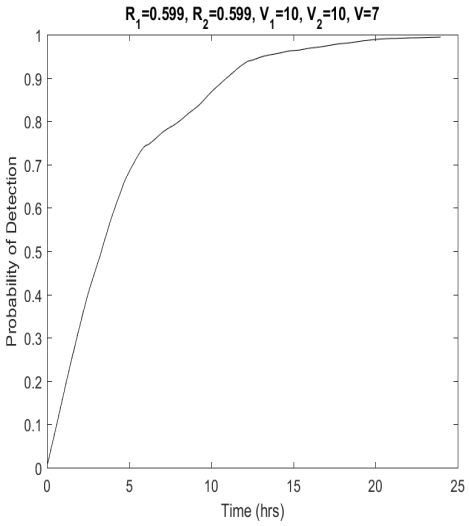
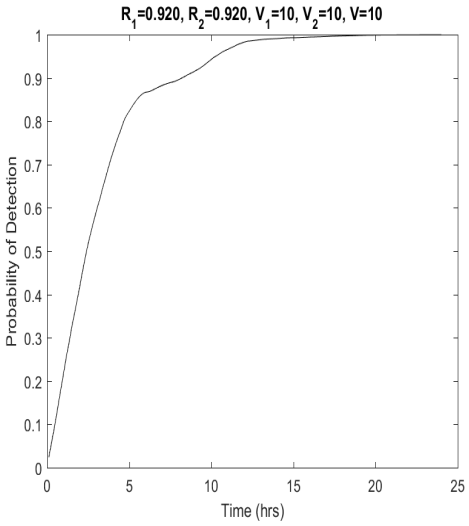


Figure 4.8. P_D vs Time



Inline Ladder Search

Figure 4.9. P_D vs Time

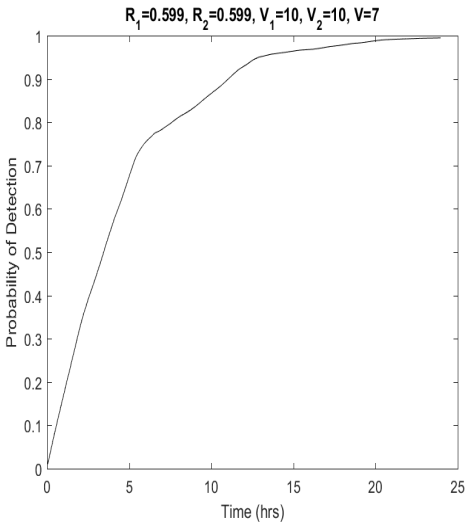
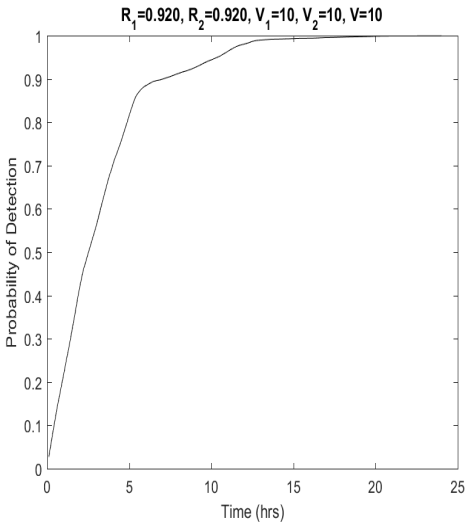


Figure 4.10. P_D vs Time



As can be seen from the figures, there is an approach to approximately $P_D = 1 - e^{-bt}$. An increase in V, as stated before increases $R(v_s)$ for a given v_s. This obviously hastens P_D to higher level more quickly.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 5:
Conclusions and Future Work

Based on the results from Chapters 2-4 the following sections summarize the conclusions drawn.

5.1 Results and Conclusions

5.1.1 Discrete and Bayesian Searches

Both the Discrete and Bayesian searches follow the greedy algorithm and give similar results. In fact, if $p_0 = 0$, then the results will be identical to within rounding error. As expected, the more concentrated p_i density, the higher the probability to detect P_D and the lower number of glimpse needed to reach it. This even has the capacity to override higher conditional miss probability in that same concentrated area. Both search systems provide predictable, repeatable results, but the Bayesian search provides a more useful output. By modifying the p_i at each step, not only can $p_0 > 0$, but faulty conjectures that created the p_i be identified by scrutinizing the changing p_i and increasing p_0. This provides a useful product by the algorithm at any point, the p_i have been calculated if the number of glimpses have been exceeded and can be used at a later date. For a discrete search, the output from the algorithm is just the failure to detect on previous searches.

5.1.2 Continuous Searches

As v_s increases, P_D has a non-intuitive result. It varies in a non-linear, way some times increasing or decreasing as v_s changes. Another non-intuitive result is that v_1 has a different change in P_D than v_2. The last result, an expected one, is that increasing V increases the P_D by increasing R.

5.2 Future Work

5.2.1 Discrete and Bayesian Searches

Improvements to the Discrete and Bayesian model in future work could include:

- Multiple p_i tested simultaneously updating all based on multiple conjectures
- Multiple q_i tested simultaneously updating all based on multiple exclusive sensors
- Time dependent q_i such as $q_i = e^{-t/\tau}$ as greedy algorithm not the optimal solution
- Time dependent p_i for a moving target.

5.2.2 Continuous Searches

Improvements to the continuous search in future work could include:

- Different search patterns
- Non-cookie cutter sensor with some non-uniform probability of detection
- Game theory approach with a target counter-detecting and evading.

APPENDIX A:
Figures

A.1 Discrete

Uniform Location and Uniform Search Density Results

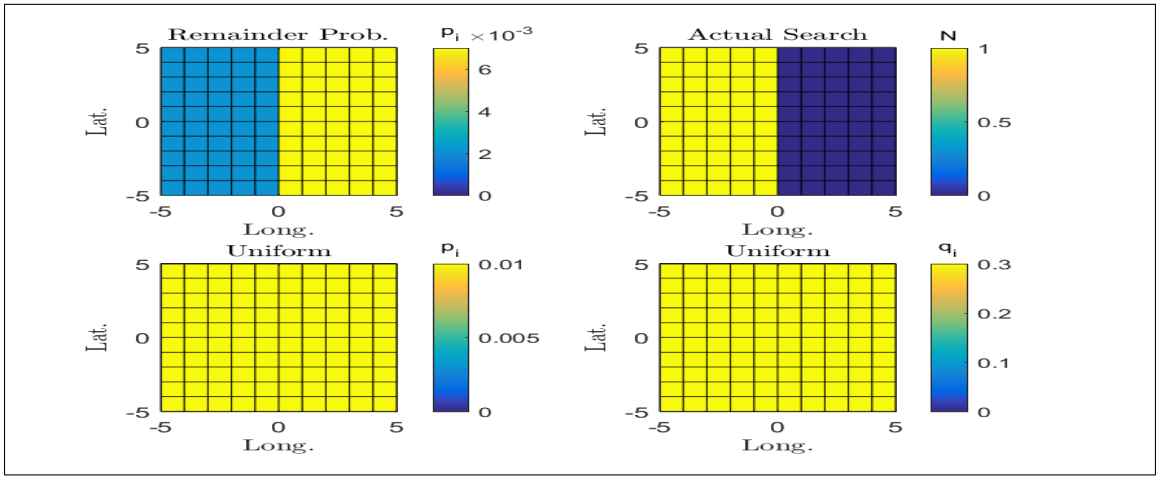


Figure A.1. Remaining probability of detection and boxes search based on uniform p_i and uniform q_i

Uniform Location and Longitudinal Stratified Search Density Results

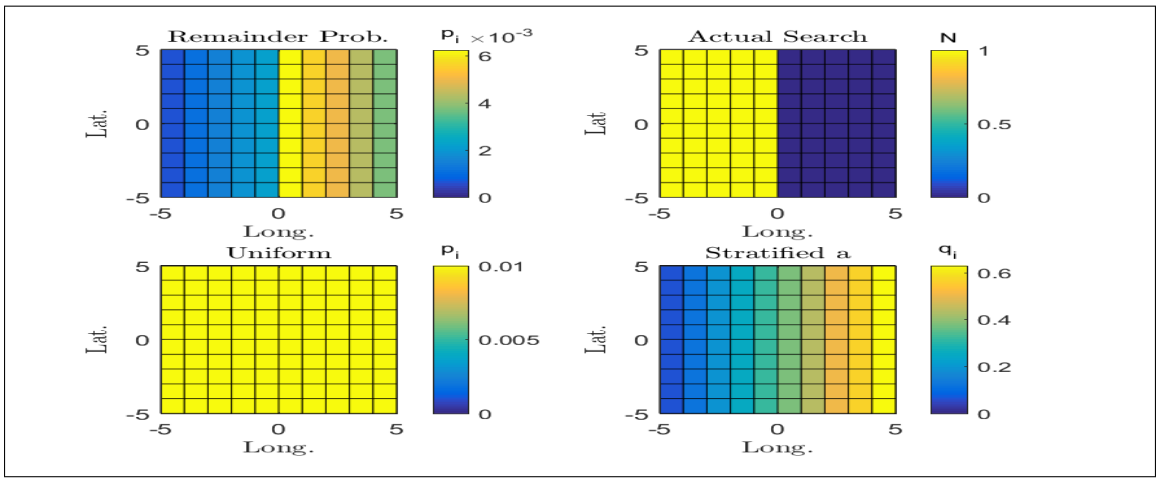


Figure A.2. Remaining probability of detection and boxes search based on uniform p_i and linear stratified q_i

Uniform Location and Corner Stratified Search Density Results

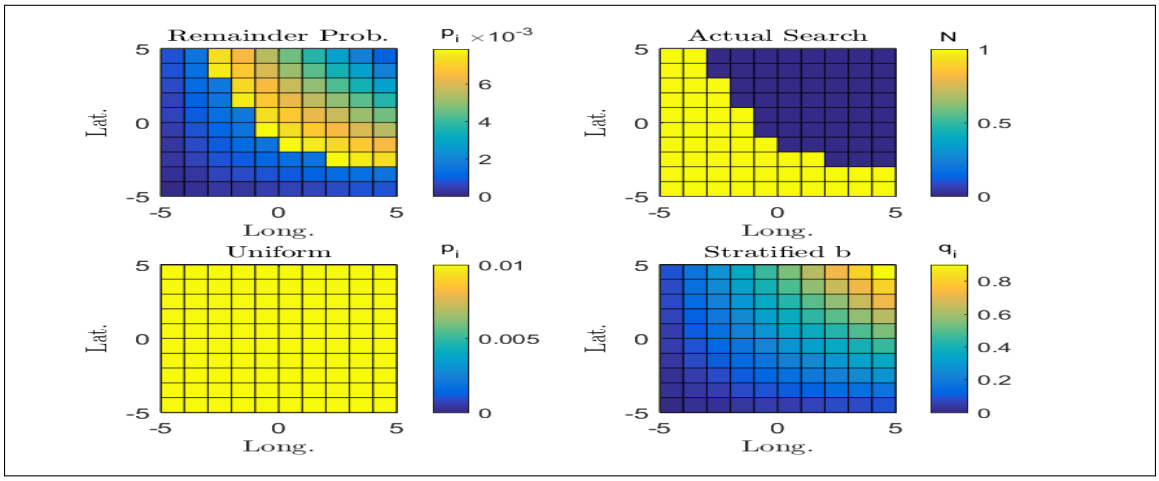


Figure A.3. Remaining probability of detection and boxes search based on uniform p_i and corner stratified q_i

Uniform Location Cumulative Probability and Fit Lines

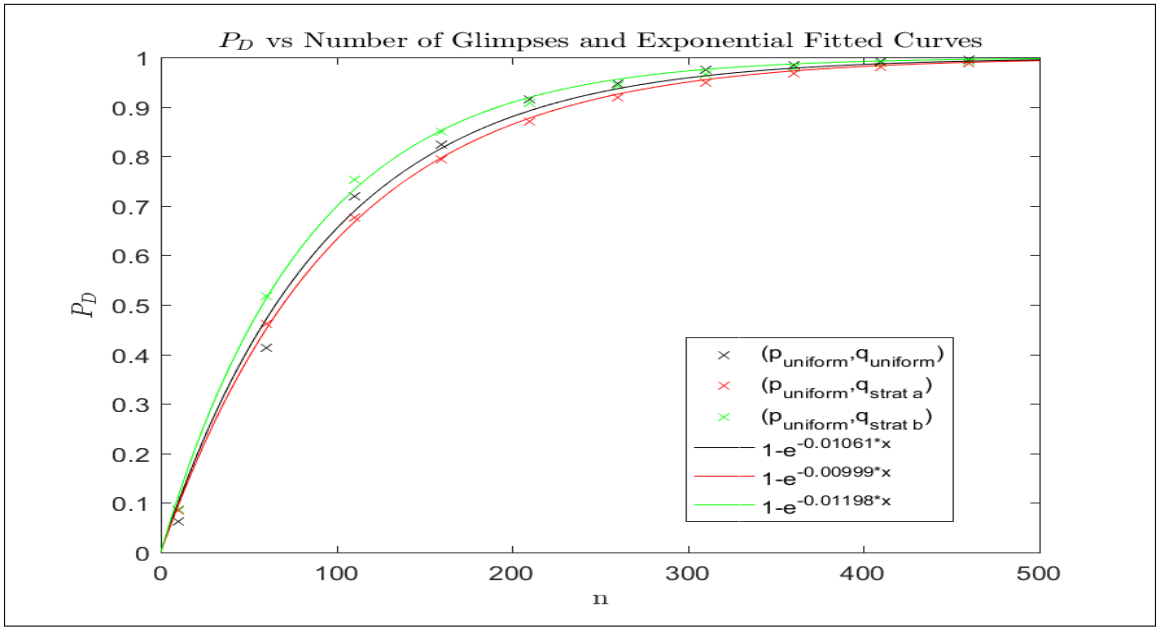


Figure A.4. Cumulative probabilities of detection based on the number of glimpses. Included are the fit best fit lines

Corner Stratified Location and Uniform Search Density Results

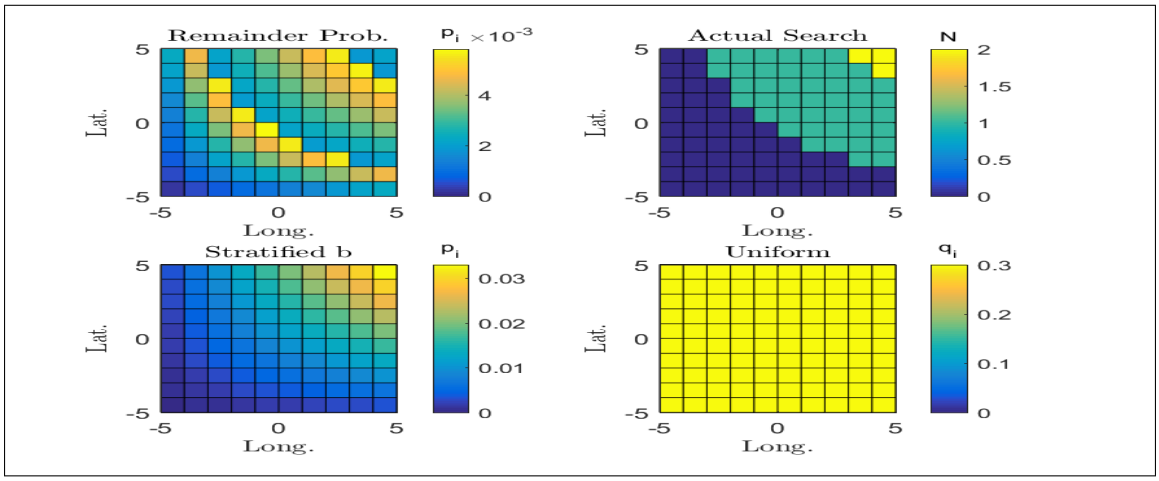


Figure A.5. Remaining probability of detection and boxes search based on corner stratified p_i and uniform q_i

Corner Stratified Location and Longitudinal Stratified Search Density Results

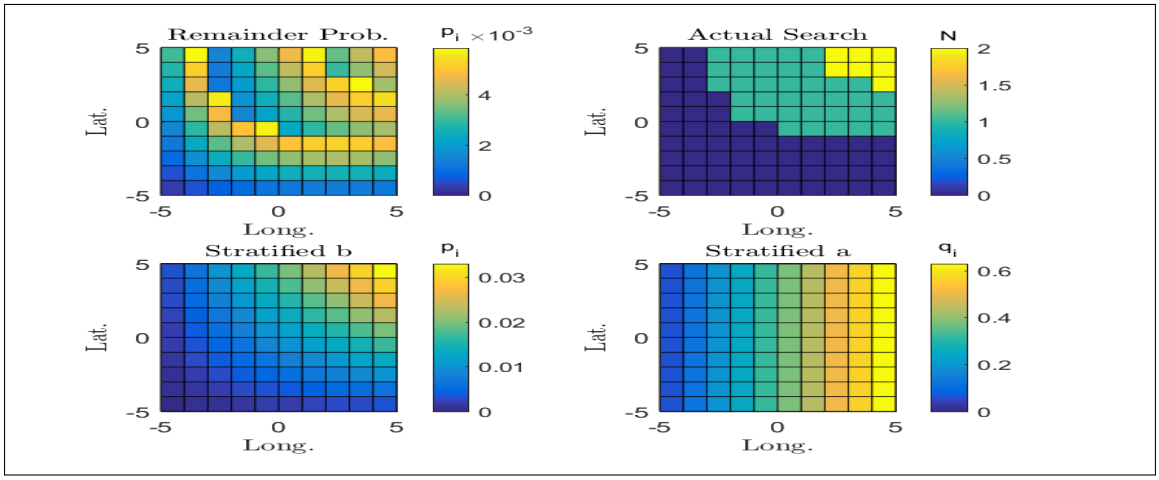


Figure A.6. Remaining probability of detection and boxes search based on corner stratified p_i and linear stratified q_i

Corner Stratified Location and Corner Stratified Search Density Results

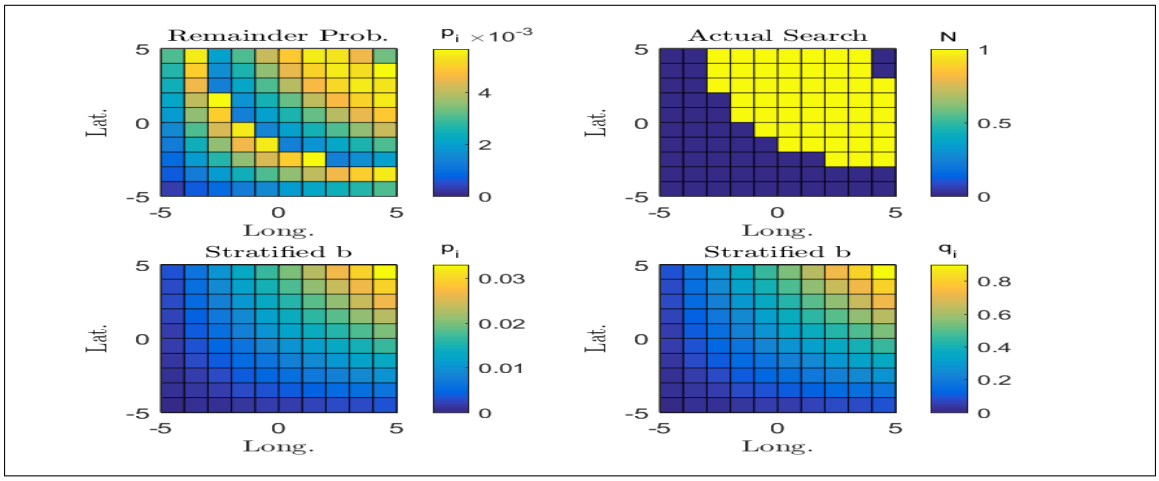


Figure A.7. Remaining probability of detection and boxes search based on corner stratified p_i and corner stratified q_i

Corner Stratified Location Cumulative Probability and Fit Lines

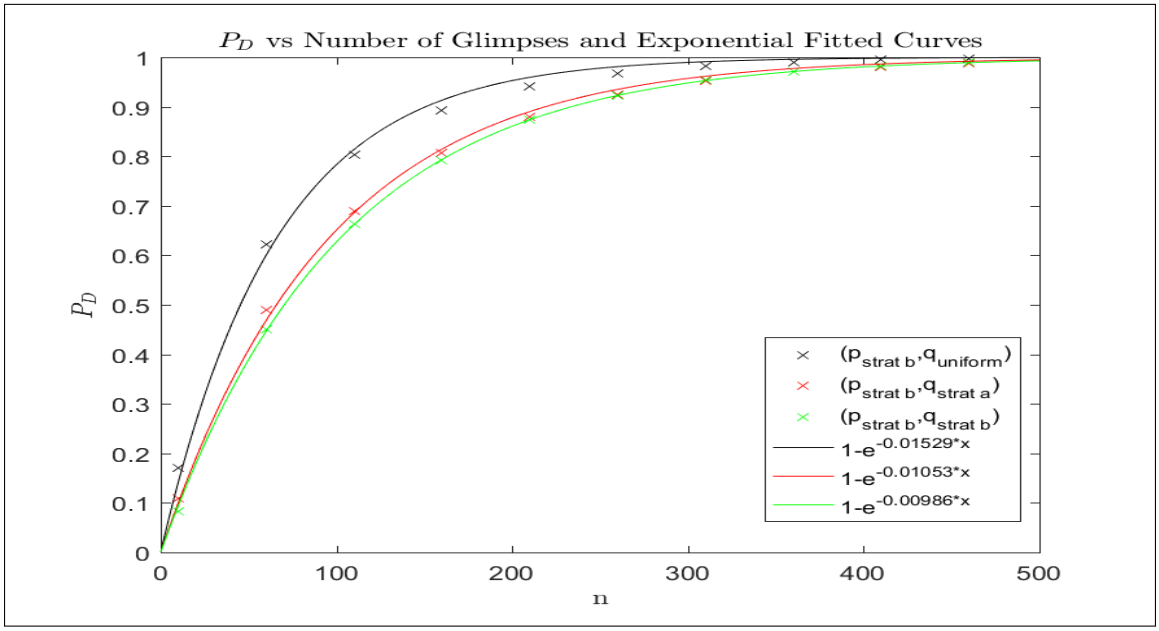


Figure A.8. Cumulative probabilities of detection based on the number of glimpses. Included are the fit best fit lines

Gaussian Location and Uniform Search Density Results

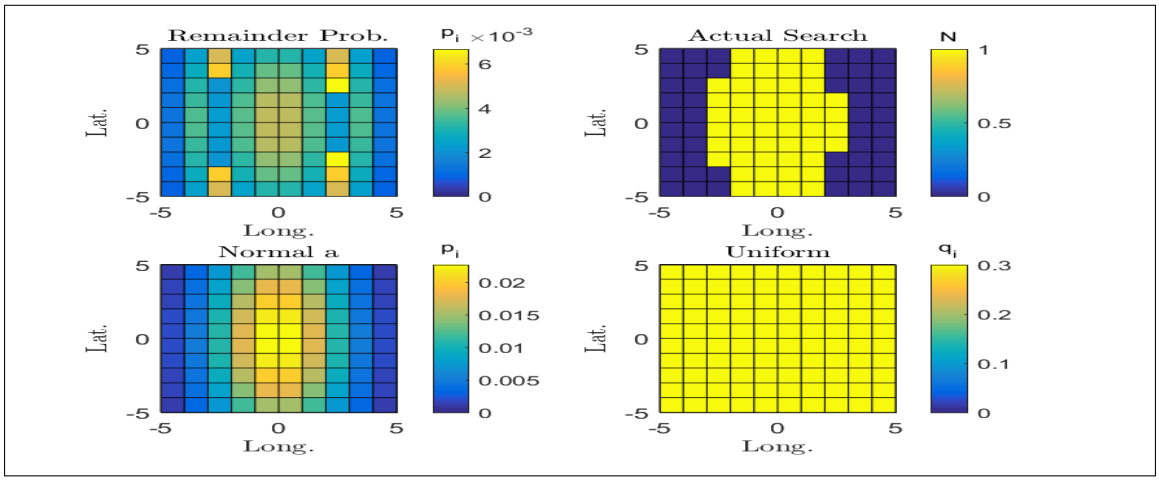


Figure A.9. Remaining probability of detection and boxes search based on Gaussian p_i and uniform q_i

Gaussian Location and Longitudinal Stratified Search Density Results

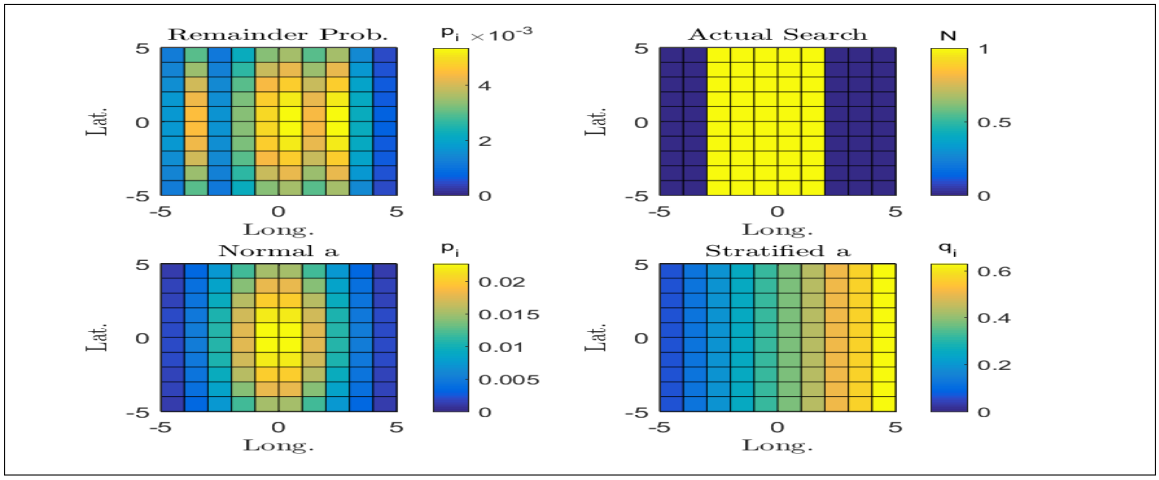


Figure A.10. Remaining probability of detection and boxes search based on Gaussian p_i and linear stratified q_i

Gaussian Location and Corner Stratified Search Density Results

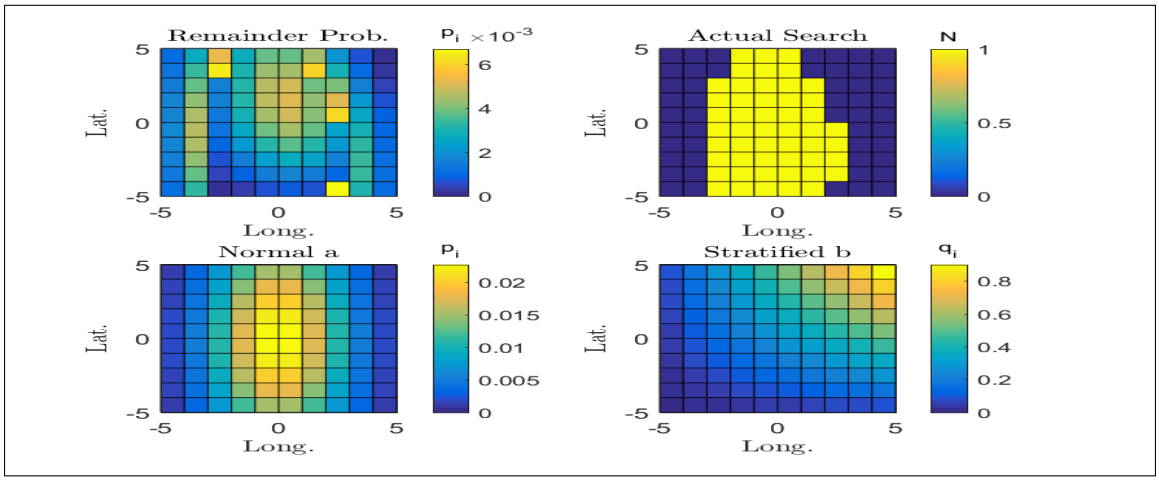


Figure A.11. Remaining probability of detection and boxes search based on Gaussian p_i and corner stratified q_i

Gaussian Location Cumulative Probability and Fit Lines

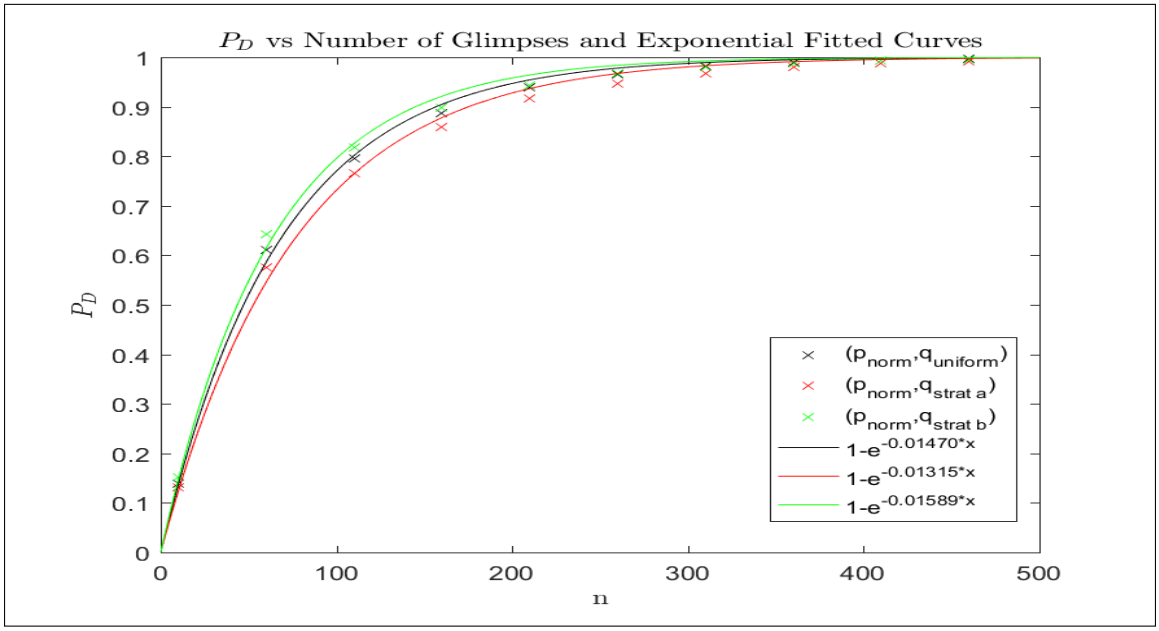


Figure A.12. Cumulative probabilities of detection based on the number of glimpses. Included are the fit best fit lines

Side Gaussian Location and Uniform Search Density Results

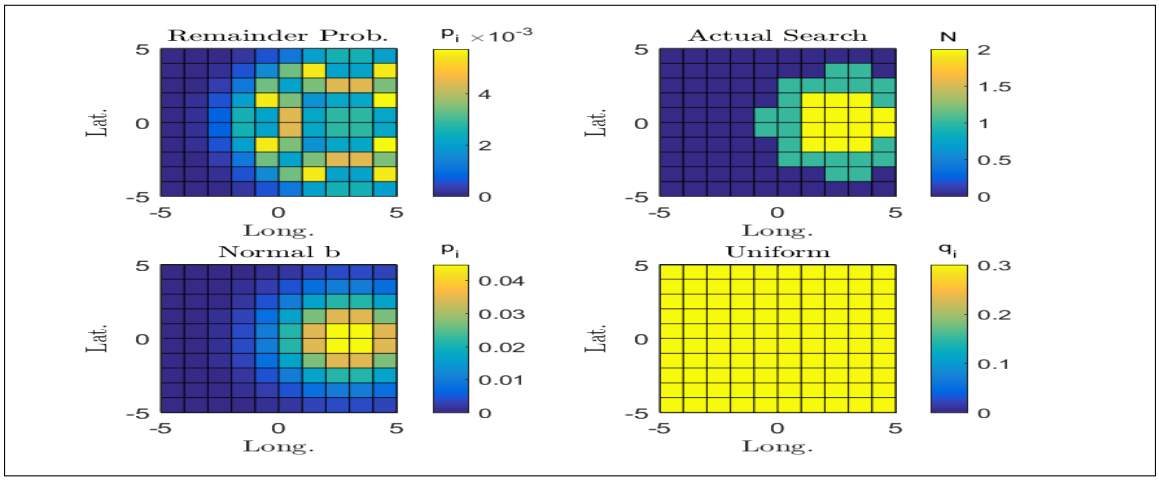


Figure A.13. Remaining probability of detection and boxes search based on side Gaussian p_i and uniform q_i

Side Gaussian Location and Longitudinal Stratified Search Density Results

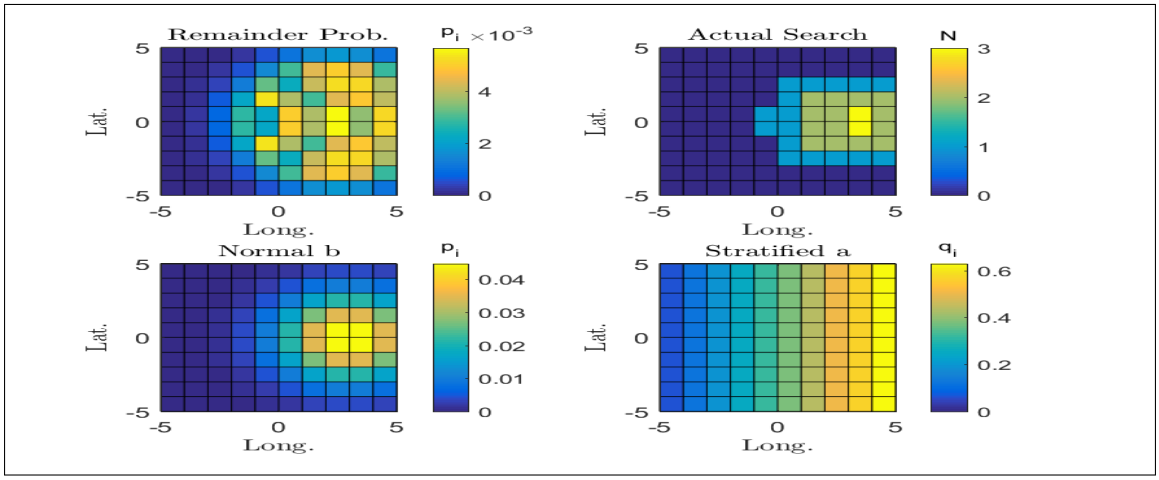


Figure A.14. Remaining probability of detection and boxes search based on side Gaussian p_i and linear stratified q_i

Side Gaussian Location and Corner Stratified Search Density Results

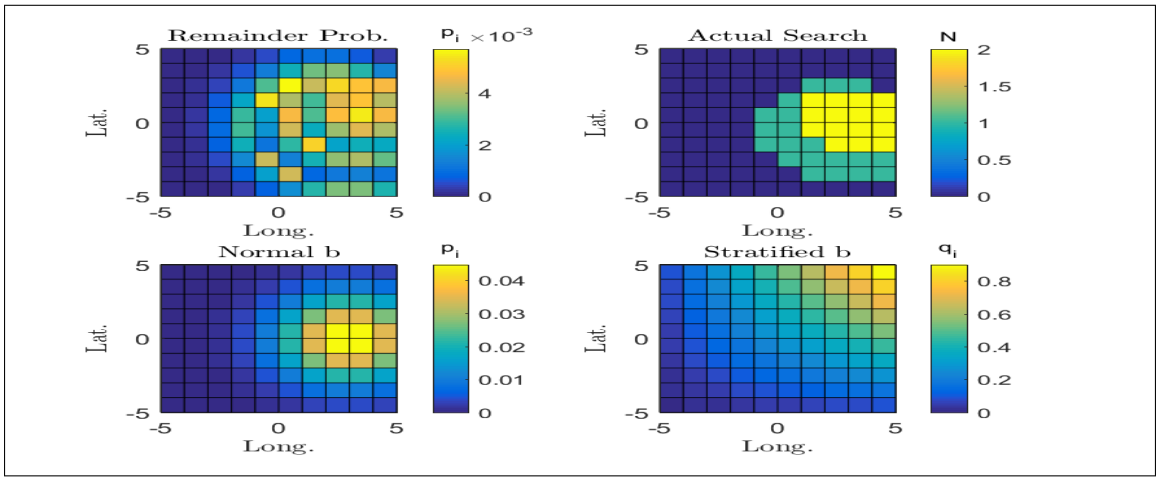


Figure A.15. Remaining probability of detection and boxes search based on side Gaussian p_i and corner stratified q_i

Side Gaussian Location Cumulative Probability and Fit Lines

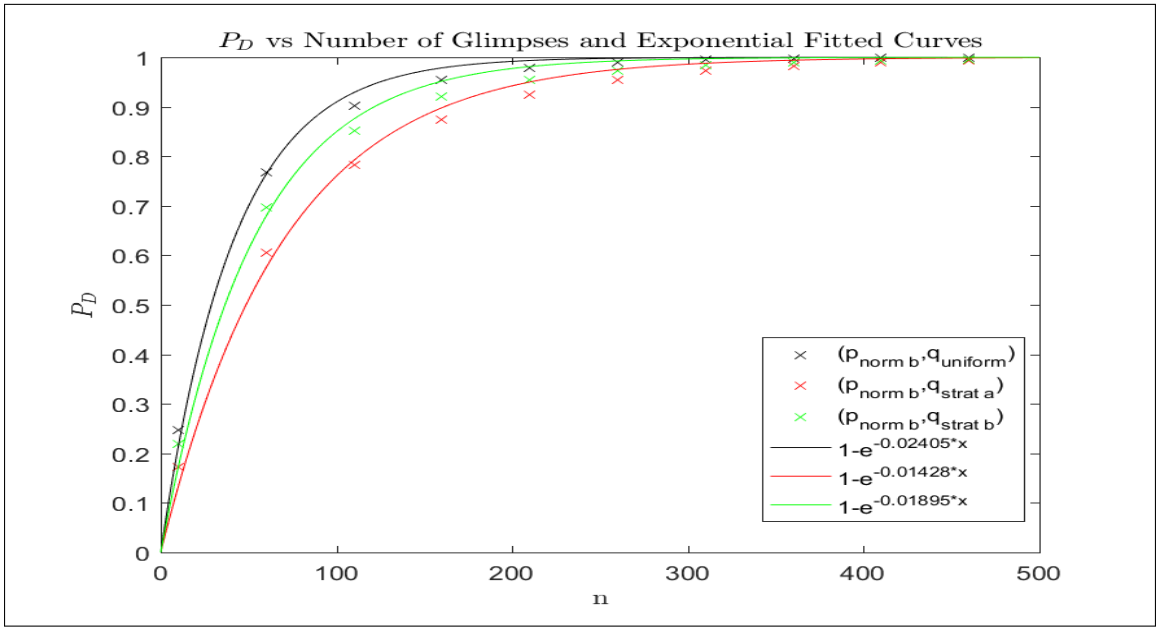


Figure A.16. Cumulative probabilities of detection based on the number of glimpses. Included are the fit best fit lines

A.2 Bayesian

Uniform Location and Uniform Search Density Results

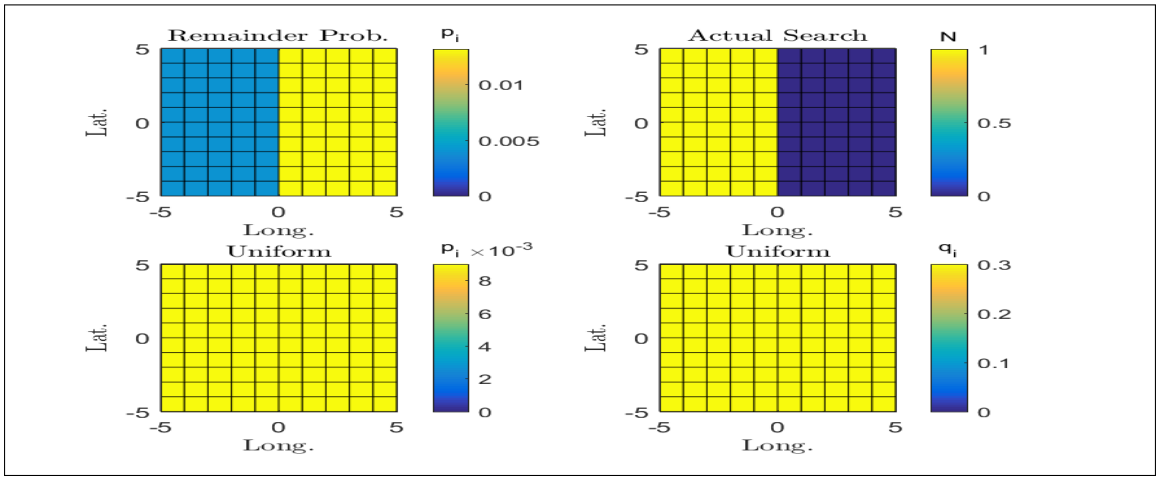


Figure A.17. Remaining probability of detection and boxes search based on uniform p_i and uniform q_i

Uniform Location and Longitudinal Stratified Search Density Results

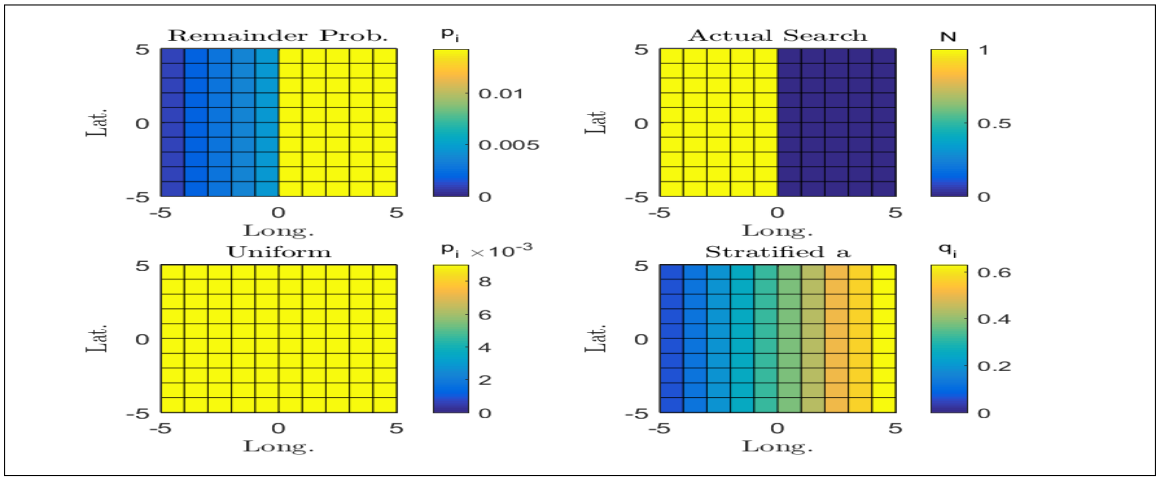


Figure A.18. Remaining probability of detection and boxes search based on uniform p_i and linear stratified q_i

Uniform Location and Corner Stratified Search Density Results

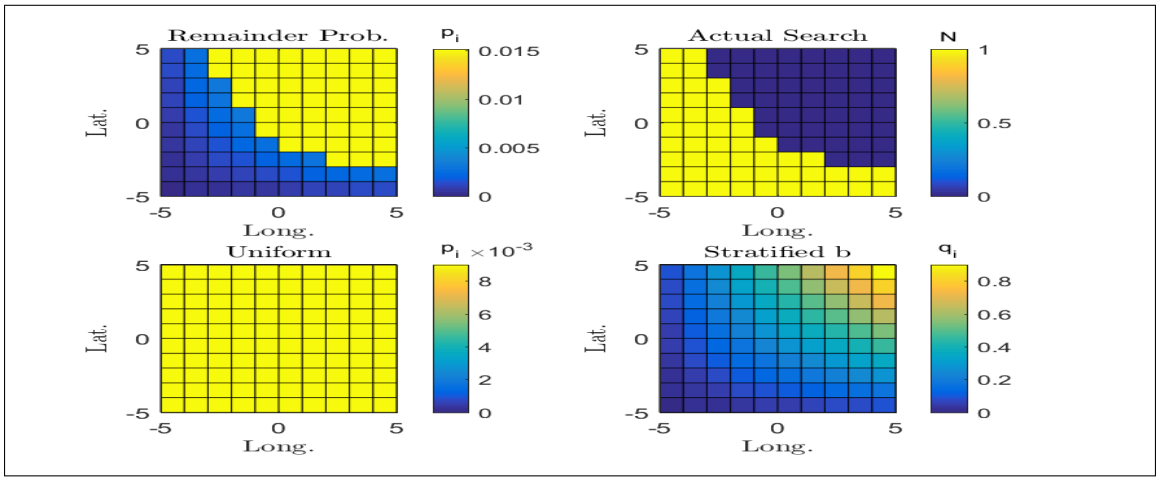


Figure A.19. Remaining probability of detection and boxes search based on uniform p_i and corner stratified q_i

Uniform Location Cumulative Probability and Fit Lines

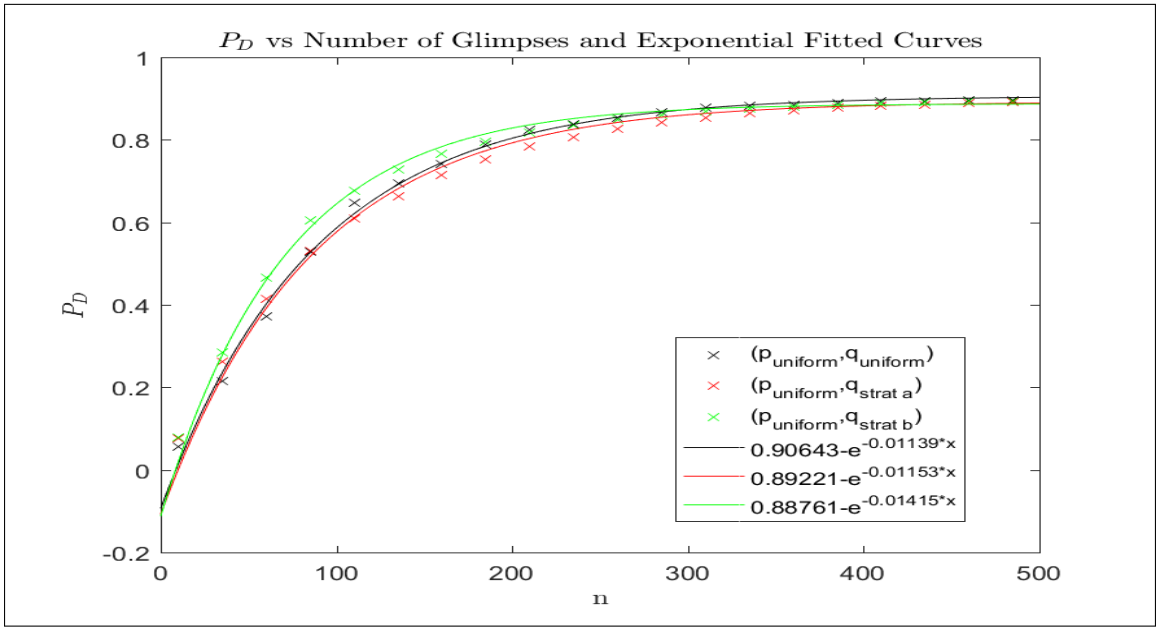


Figure A.20. Cumulative probabilities of detection based on the number of glimpses. Included are the fit best fit lines

Corner Stratified Location and Uniform Search Density Results

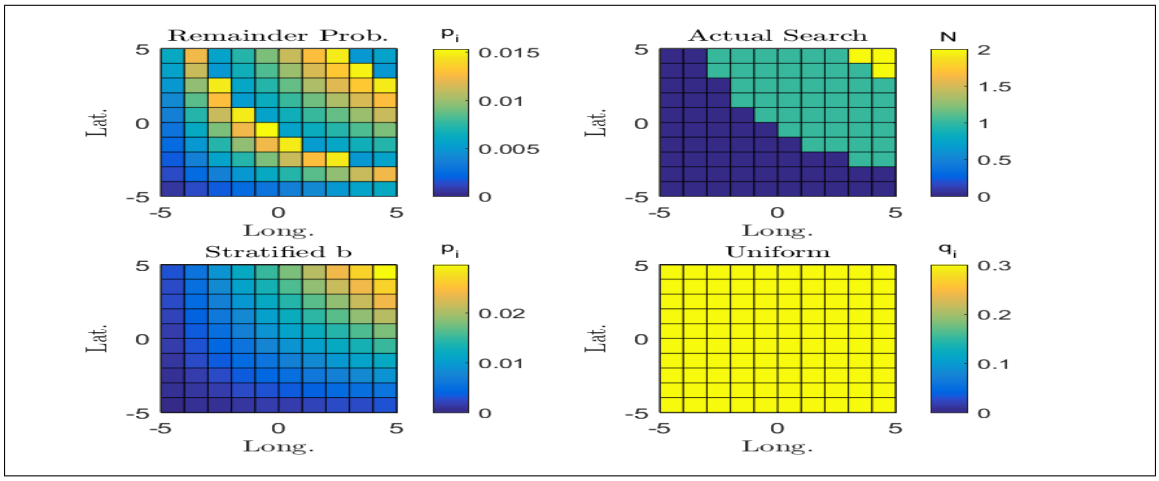


Figure A.21. Remaining probability of detection and boxes search based on corner stratified p_i and uniform q_i

Corner Stratified Location and Longitudinal Stratified Search Density Results

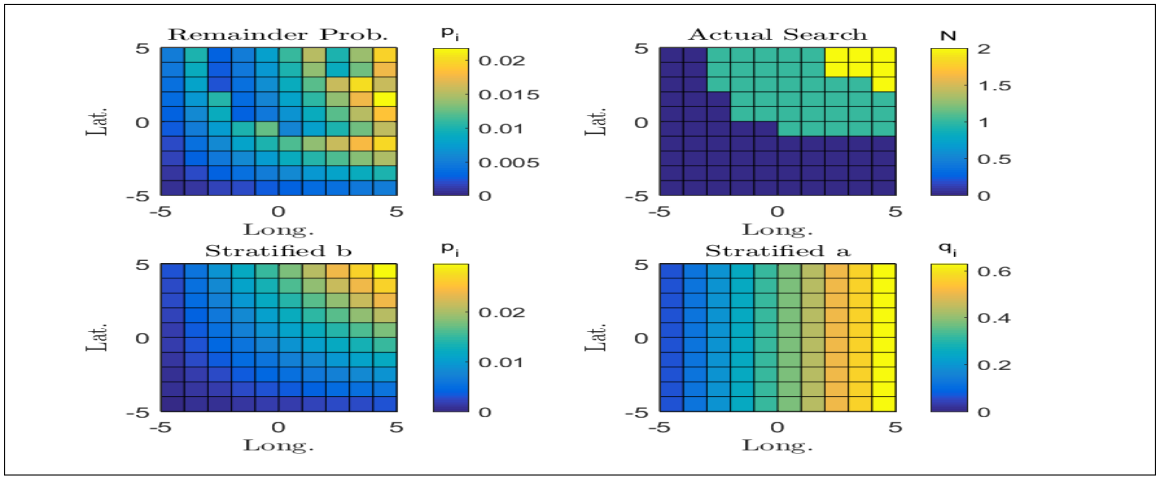


Figure A.22. Remaining probability of detection and boxes search based on corner stratified p_i and linear stratified q_i

Corner Stratified Location and Corner Stratified Search Density Results

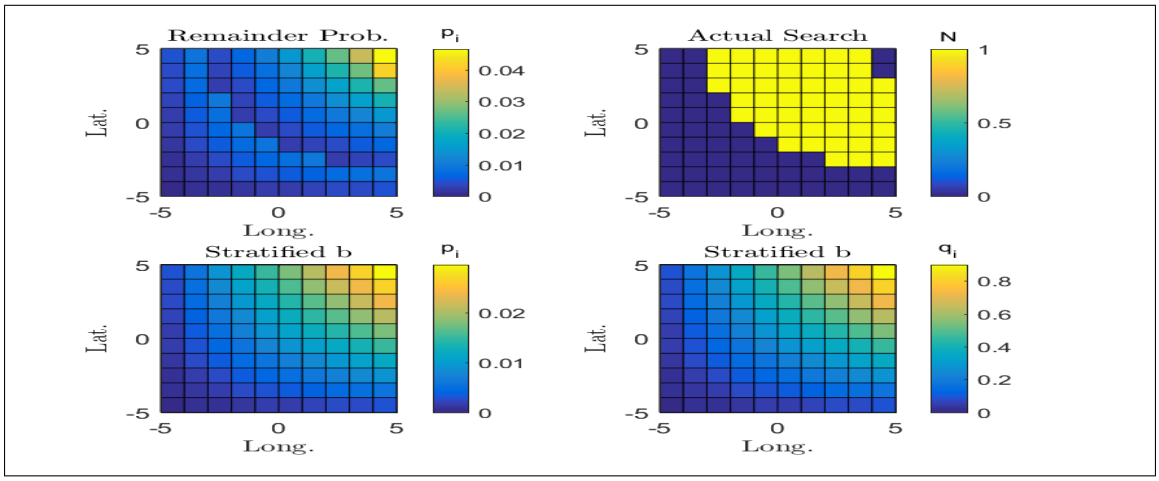


Figure A.23. Remaining probability of detection and boxes search based on corner stratified p_i and corner stratified q_i

Corner Stratified Location Cumulative Probability and Fit Lines

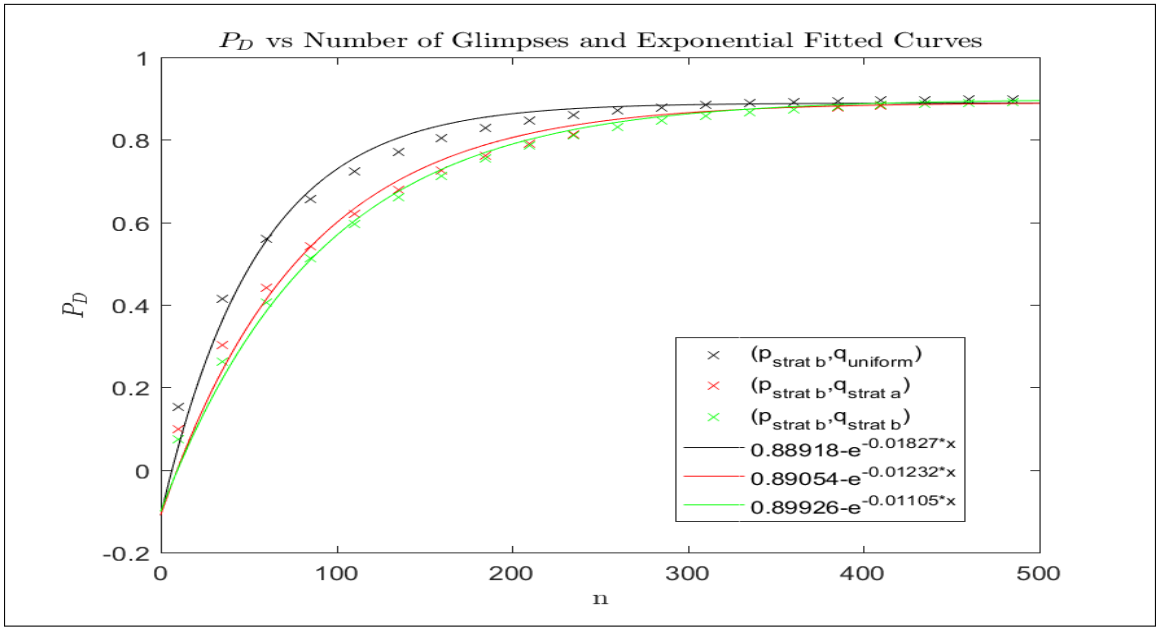


Figure A.24. Cumulative probabilities of detection based on the number of glimpses. Included are the fit best fit lines

Gaussian Location and Uniform Search Density Results

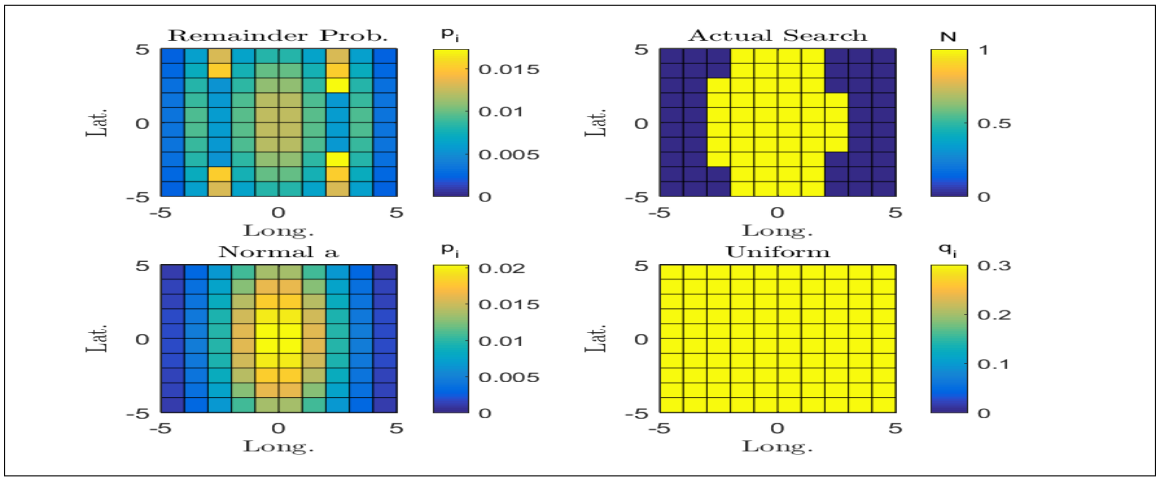


Figure A.25. Remaining probability of detection and boxes search based on Gaussian p_i and uniform q_i

Gaussian Location and Longitudinal Stratified Search Density Results

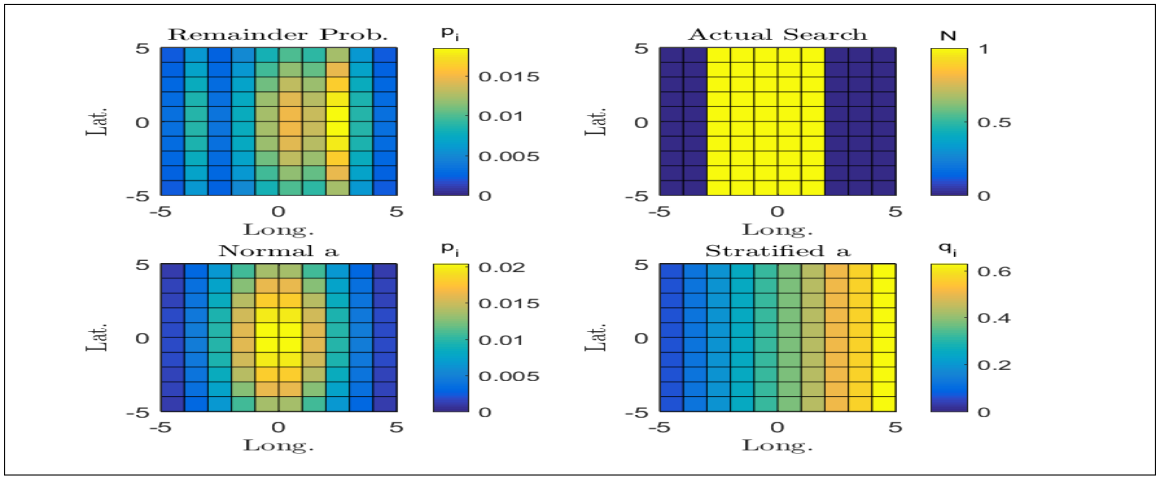


Figure A.26. Remaining probability of detection and boxes search based on Gaussian p_i and linear stratified q_i

Gaussian Location and Corner Stratified Search Density Results

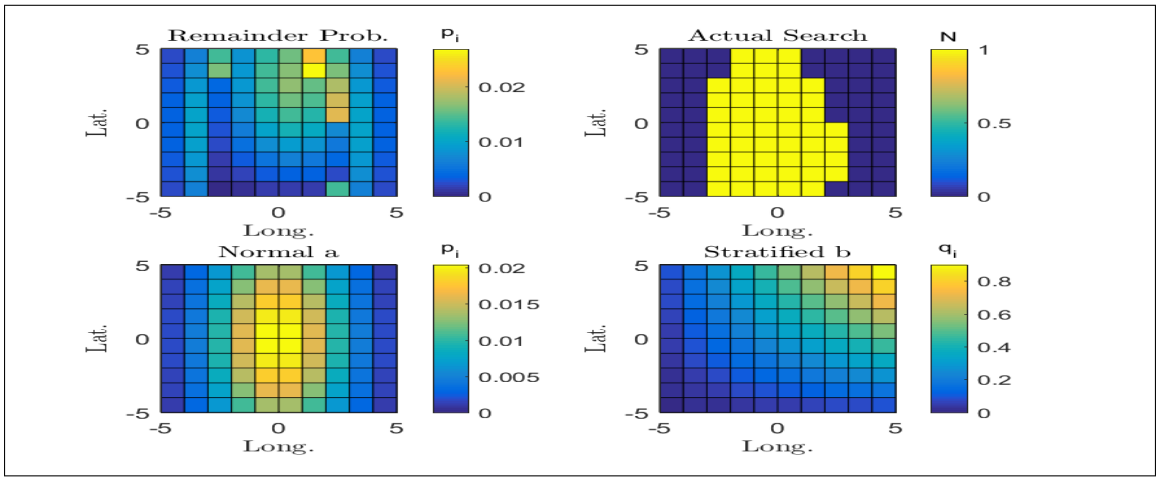


Figure A.27. Remaining probability of detection and boxes search based on Gaussian p_i and corner stratified q_i

Gaussian Location Cumulative Probability and Fit Lines

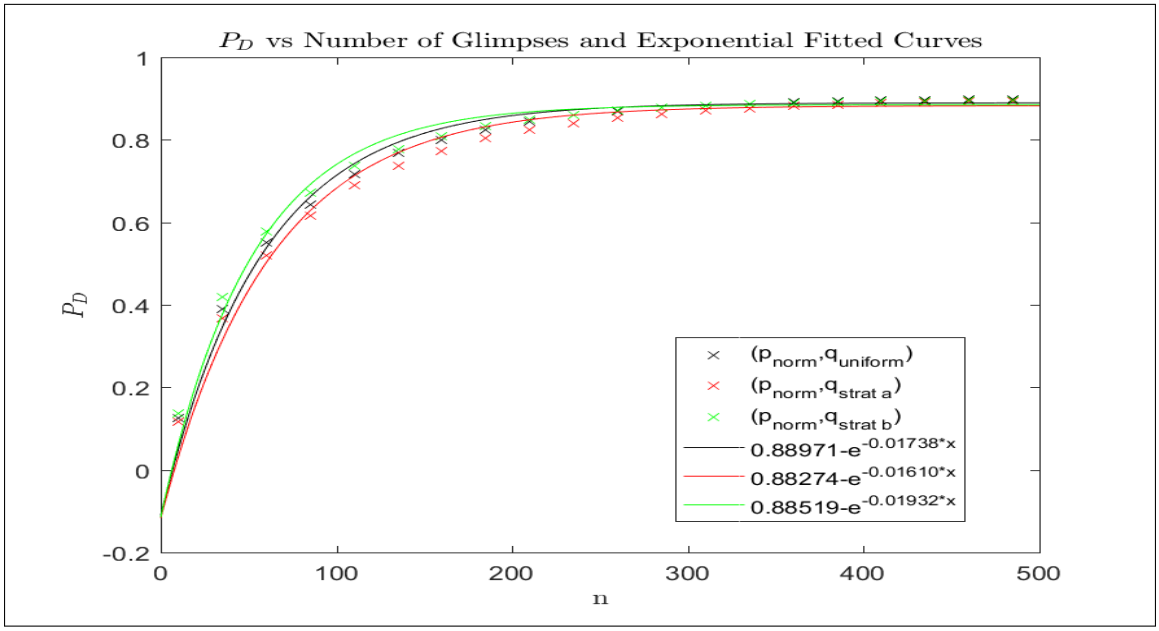


Figure A.28. Cumulative probabilities of detection based on the number of glimpses. Included are the fit best fit lines

Side Gaussian Location and Uniform Search Density Results

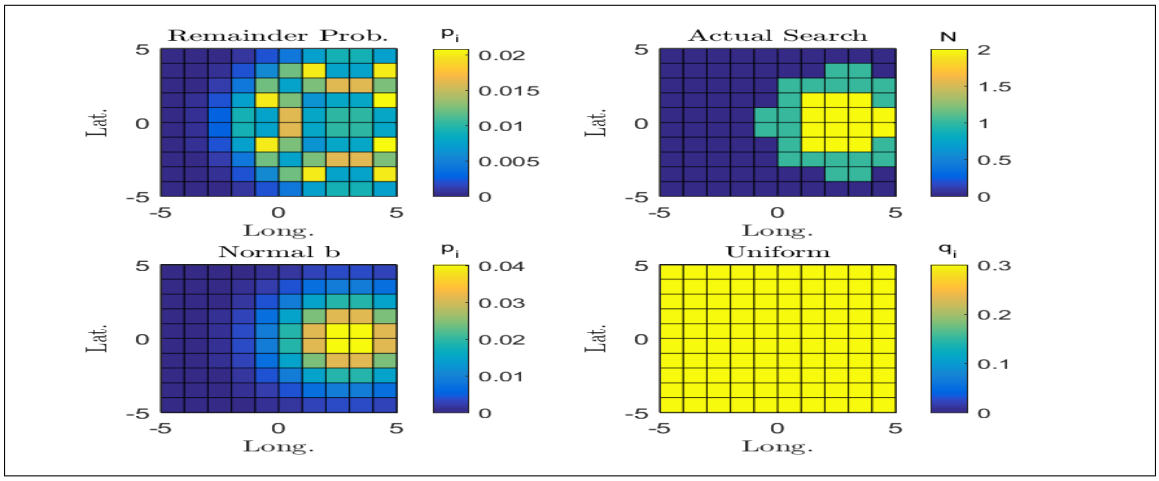


Figure A.29. Remaining probability of detection and boxes search based on side Gaussian p_i and uniform q_i

Side Gaussian Location and Longitudinal Stratified Search Density Results

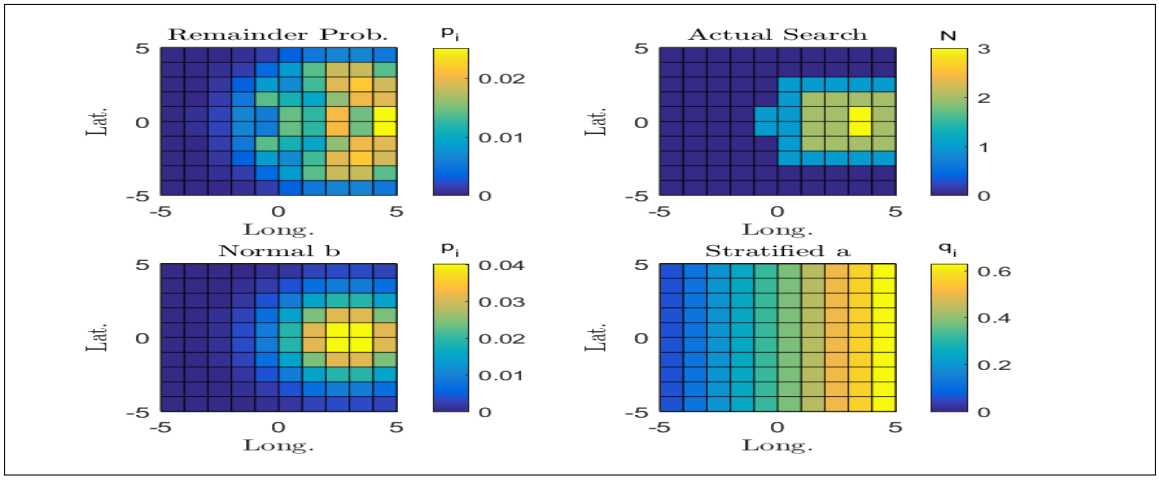


Figure A.30. Remaining probability of detection and boxes search based on side Gaussian p_i and linear stratified q_i

Side Gaussian Location and Corner Stratified Search Density Results

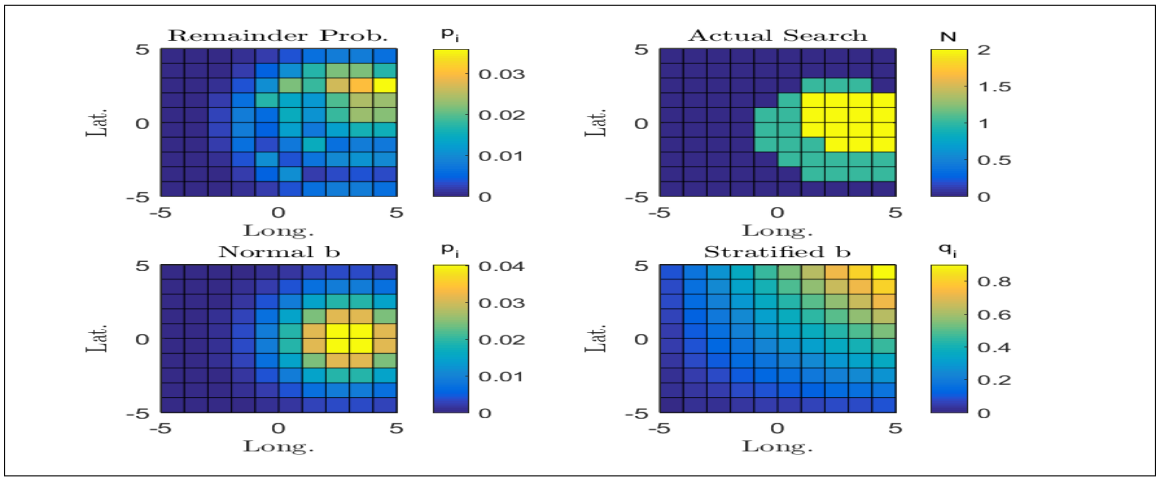


Figure A.31. Remaining probability of detection and boxes search based on side Gaussian p_i and corner stratified q_i

Side Gaussian Location Cumulative Probability and Fit Lines

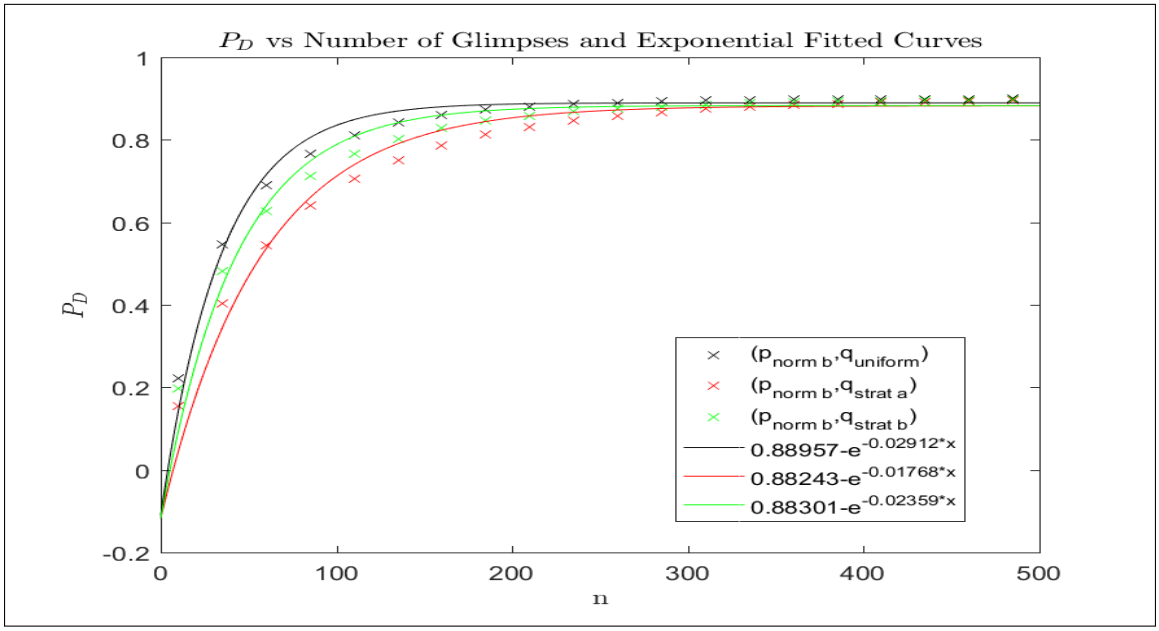


Figure A.32. Cumulative probabilities of detection based on the number of glimpses. Included are the fit best fit lines

A.3 Continuous

A.3.1 Inline Ladder Search

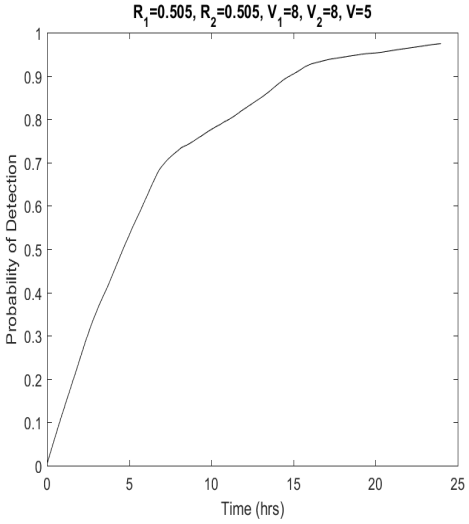


Figure A.33. P_D vs Time

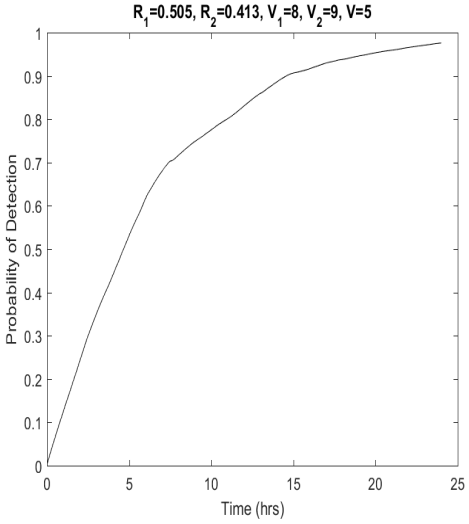


Figure A.34. P_D vs Time

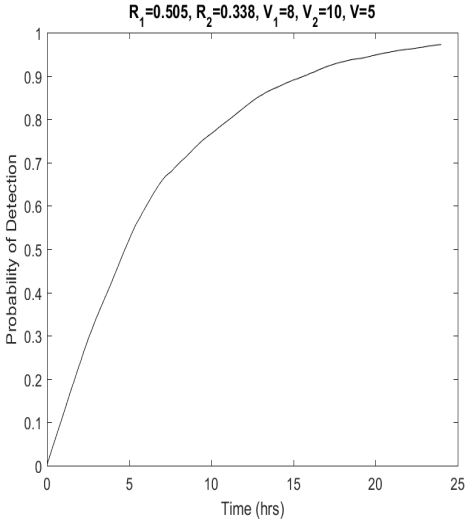


Figure A.35. P_D vs Time

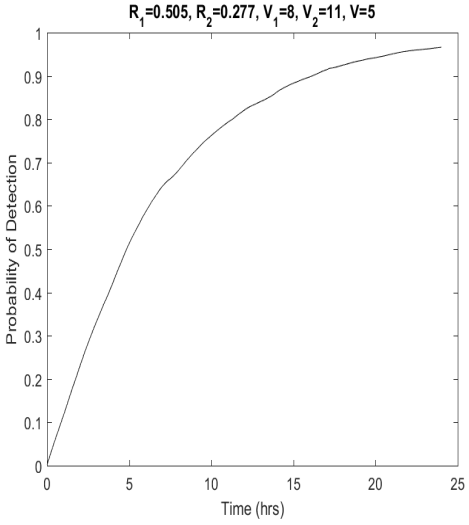


Figure A.36. P_D vs Time

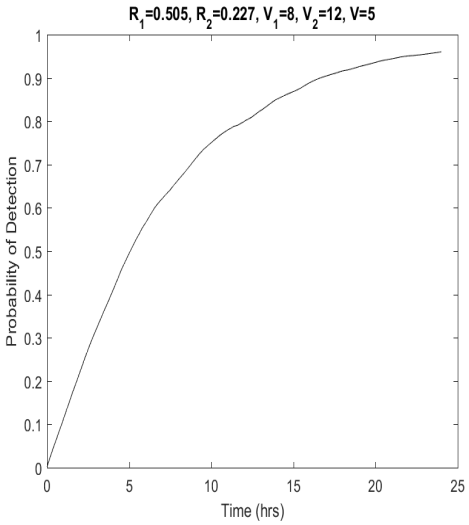


Figure A.37. P_D vs Time

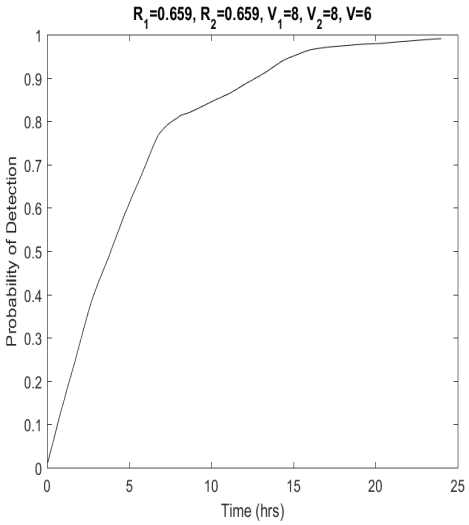


Figure A.38. P_D vs Time

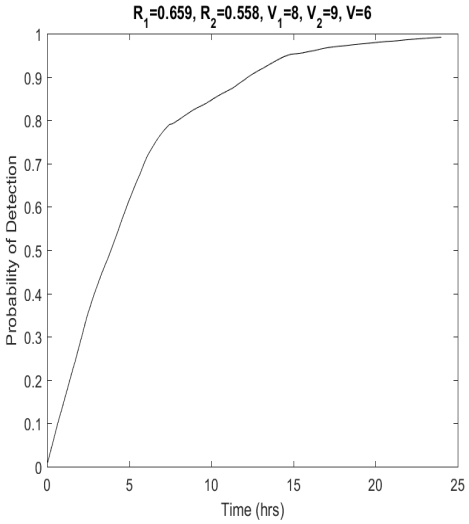


Figure A.39. P_D vs Time

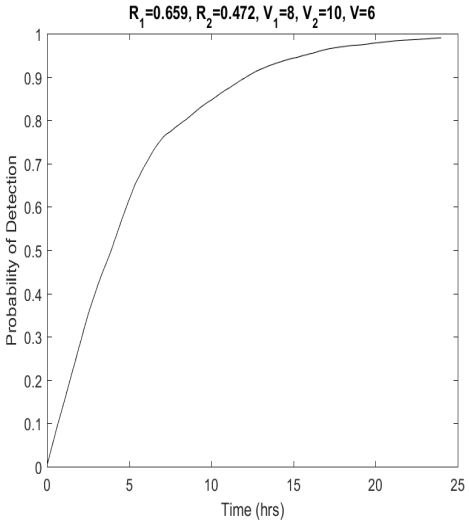


Figure A.40. P_D vs Time

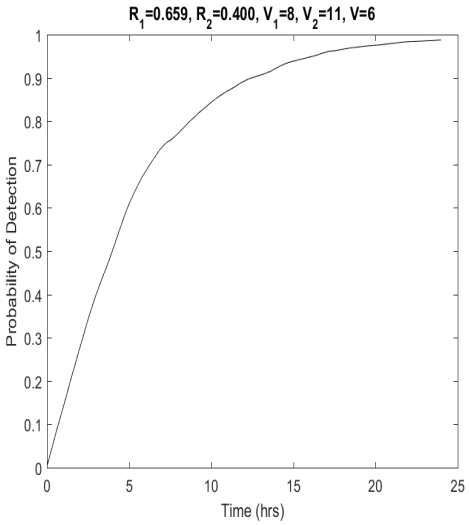


Figure A.41. P_D vs Time

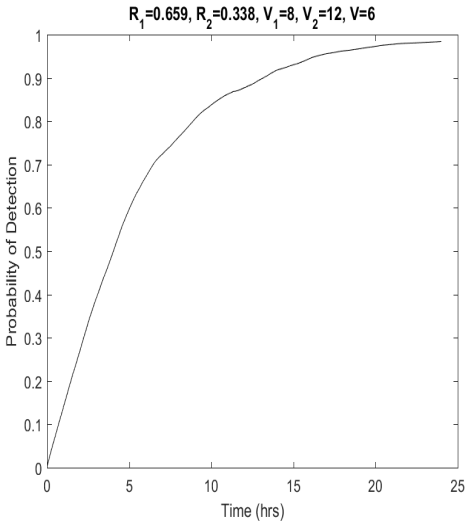


Figure A.42. P_D vs Time

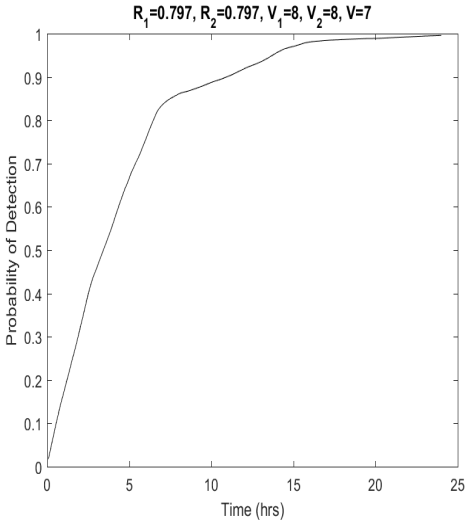


Figure A.43. P_D vs Time

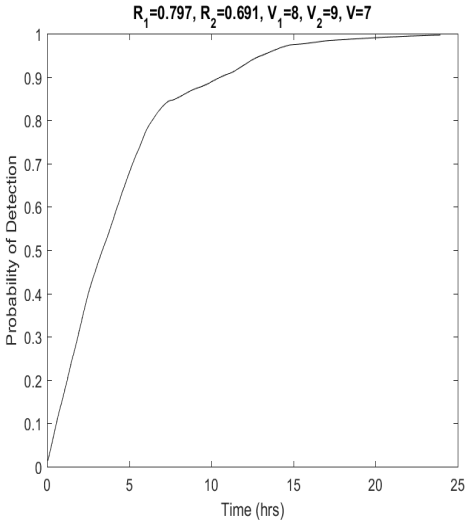


Figure A.44. P_D vs Time

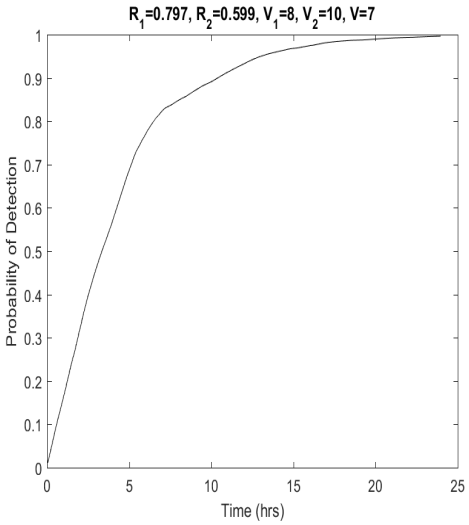


Figure A.45. P_D vs Time

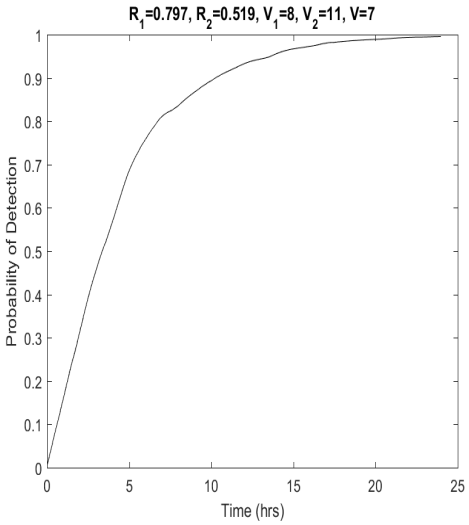


Figure A.46. P_D vs Time

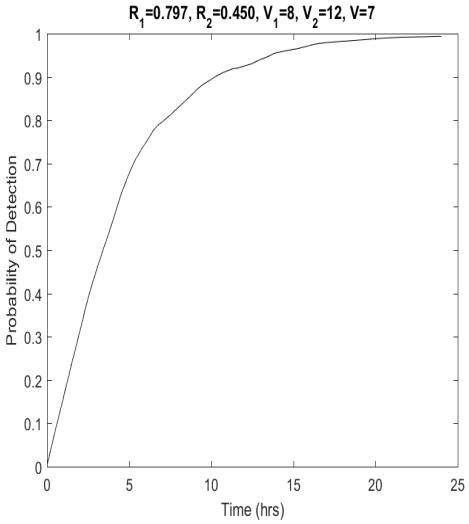


Figure A.47. P_D vs Time

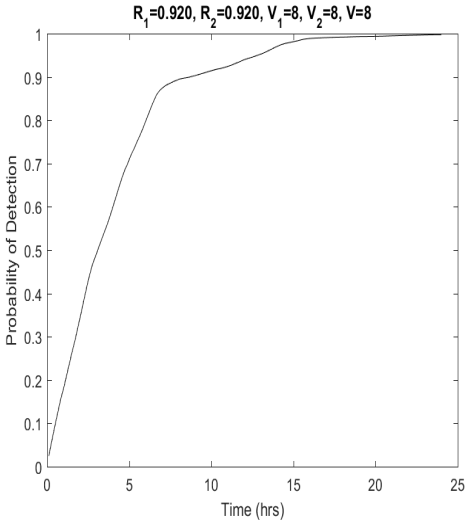


Figure A.48. P_D vs Time

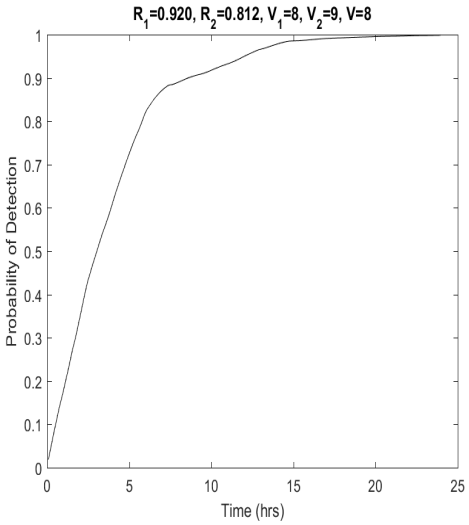


Figure A.49. P_D vs Time

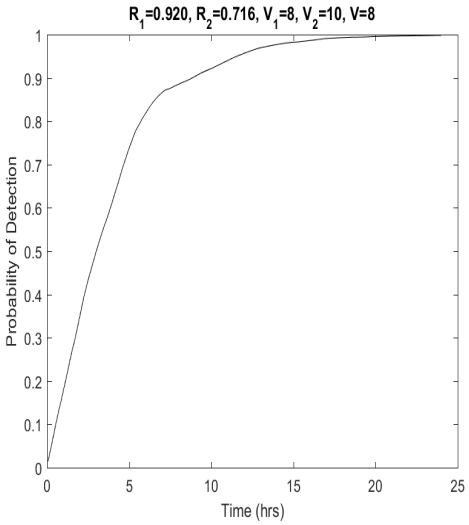


Figure A.50. P_D vs Time

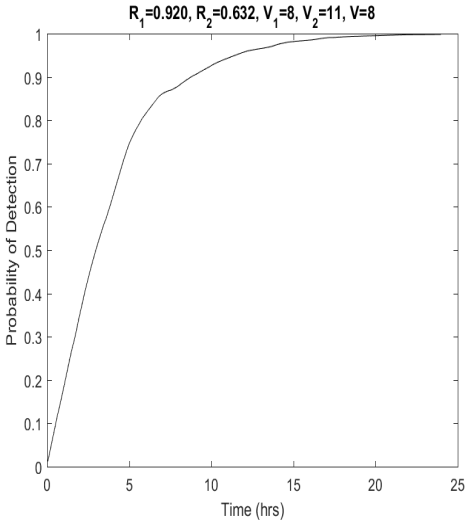


Figure A.51. P_D vs Time

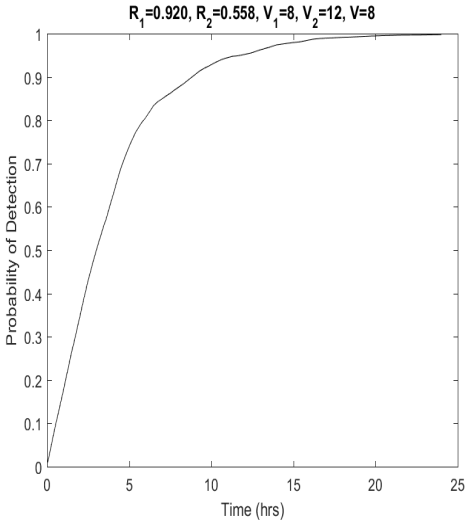


Figure A.52. P_D vs Time

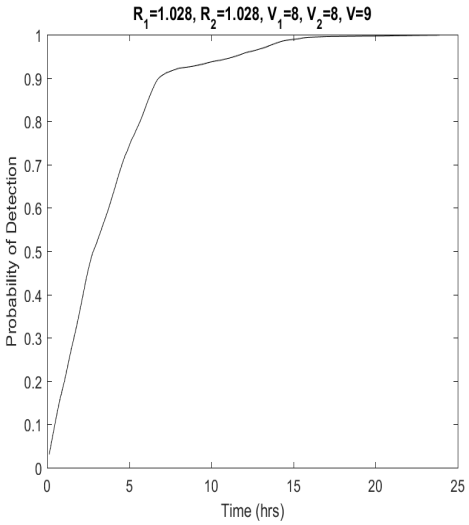


Figure A.53. P_D vs Time

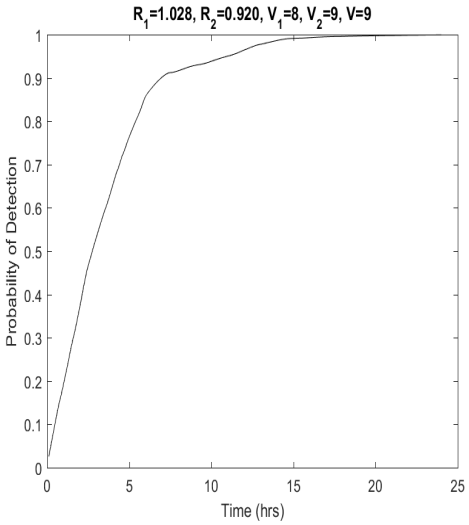


Figure A.54. P_D vs Time

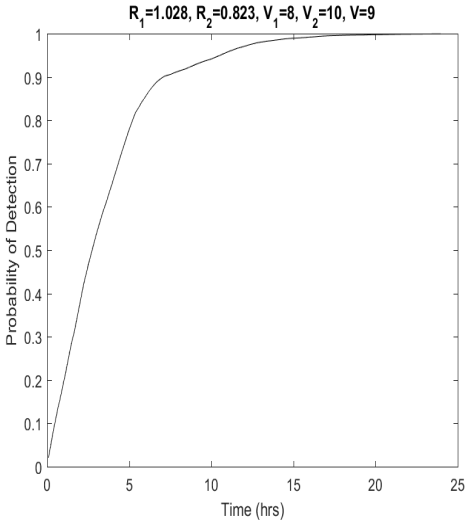


Figure A.55. P_D vs Time

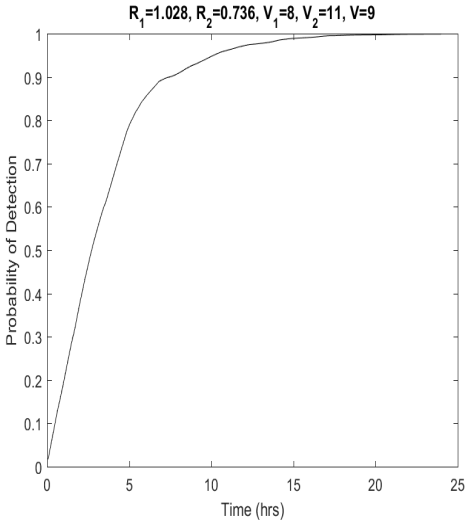


Figure A.56. P_D vs Time

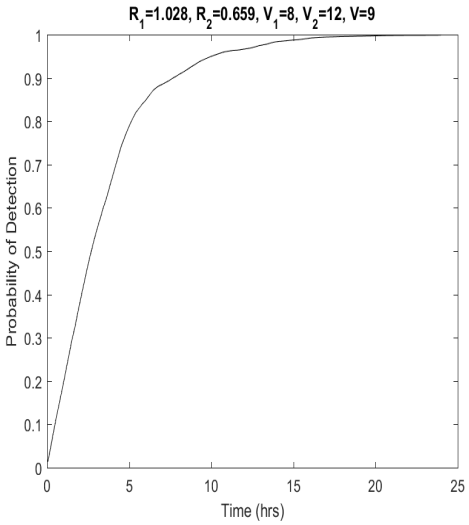


Figure A.57. P_D vs Time

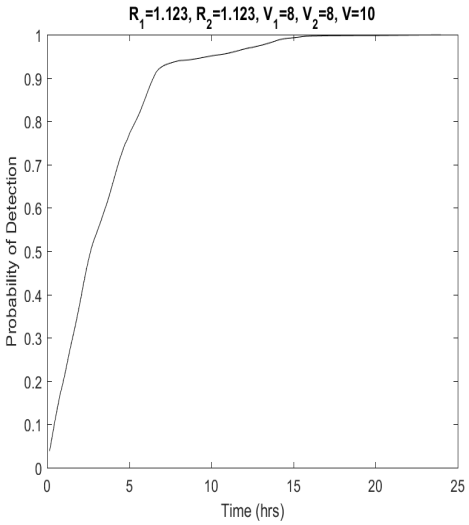


Figure A.58. P_D vs Time

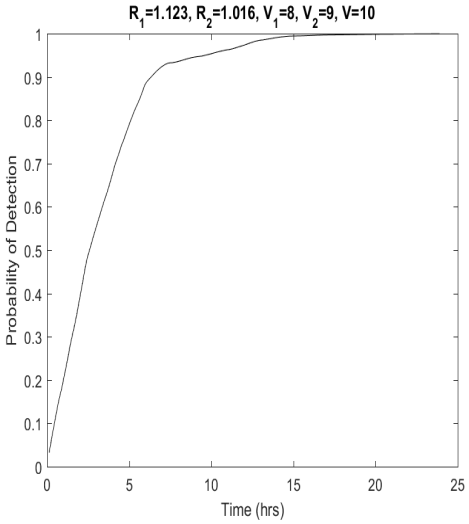


Figure A.59. P_D vs Time

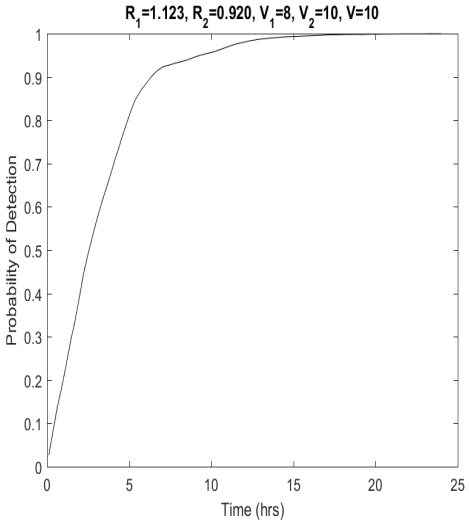


Figure A.60. P_D vs Time

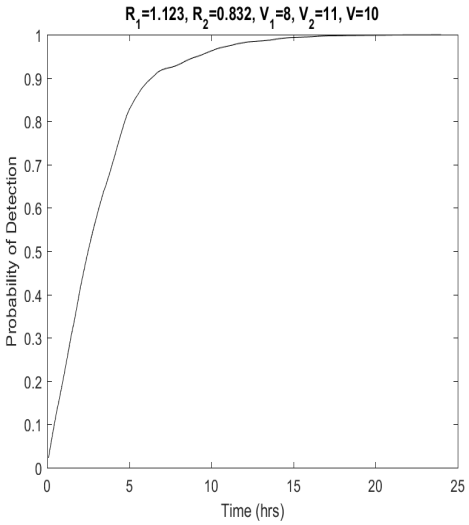


Figure A.61. P_D vs Time

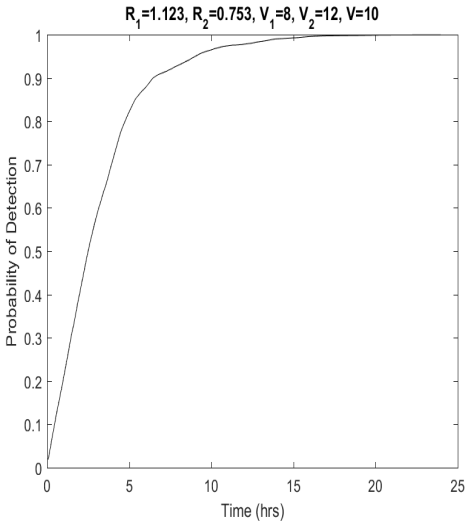


Figure A.62. P_D vs Time

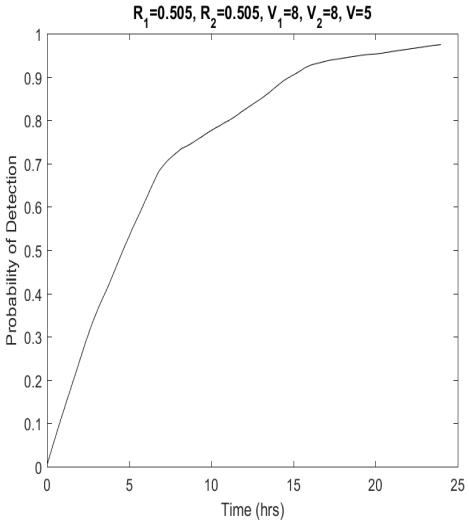


Figure A.63. P_D vs Time

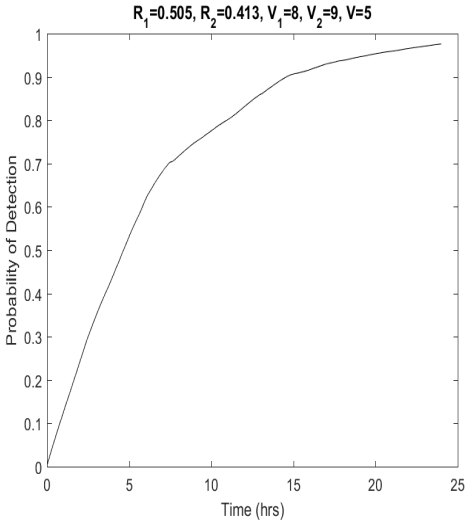


Figure A.64. P_D vs Time

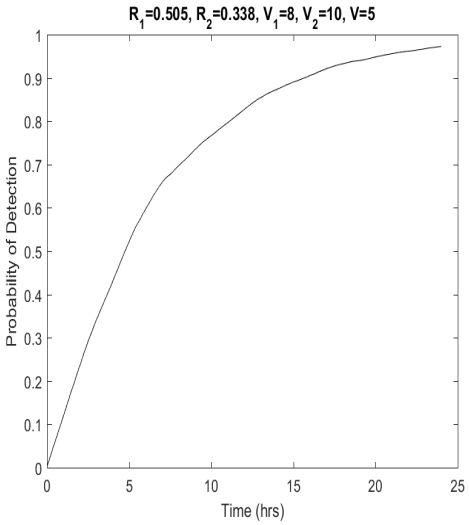


Figure A.65. P_D vs Time

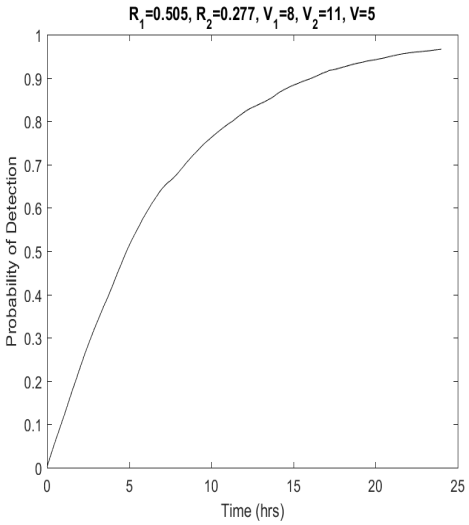


Figure A.66. P_D vs Time

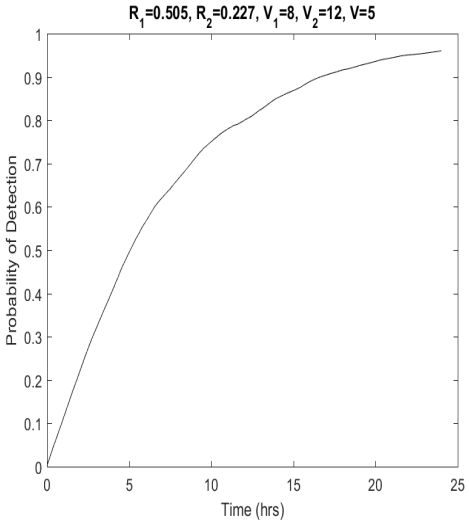


Figure A.67. P_D vs Time

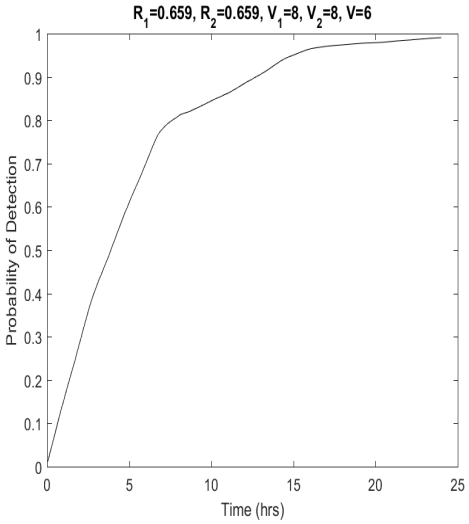


Figure A.68. P_D vs Time

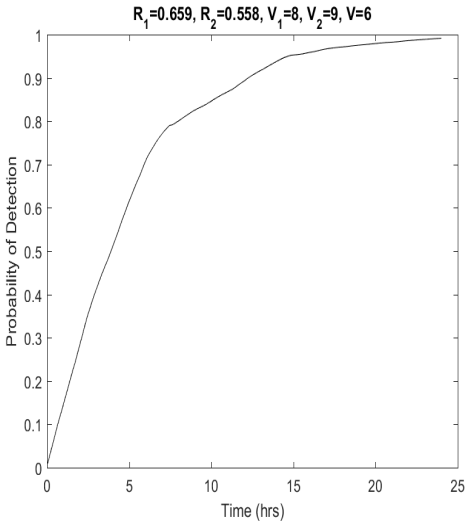


Figure A.69. P_D vs Time

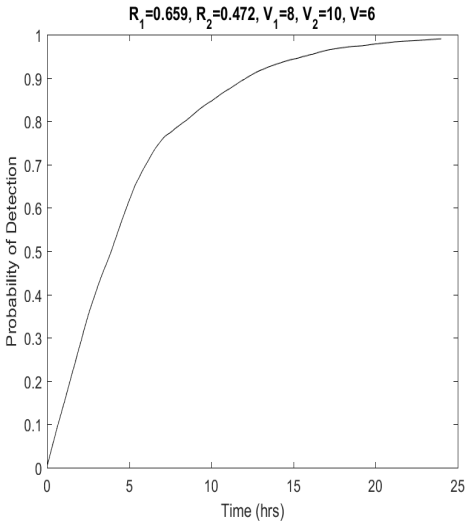


Figure A.70. P_D vs Time

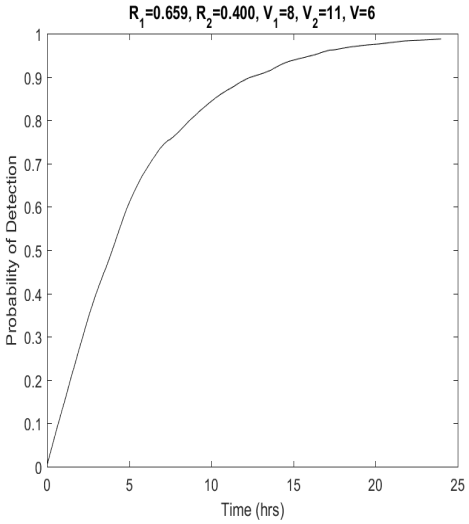


Figure A.71. P_D vs Time

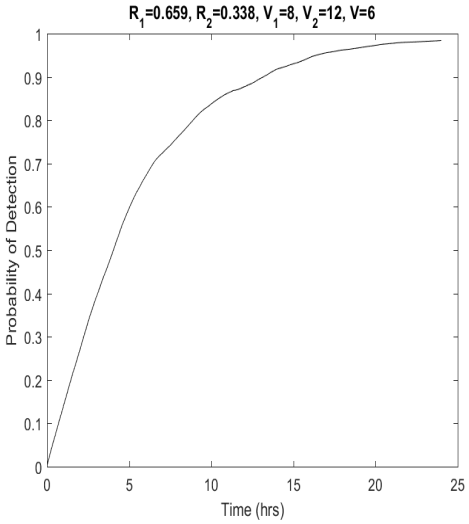


Figure A.72. P_D vs Time

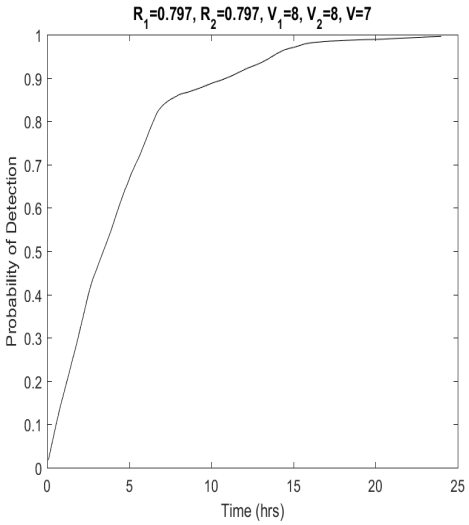


Figure A.73. P_D vs Time

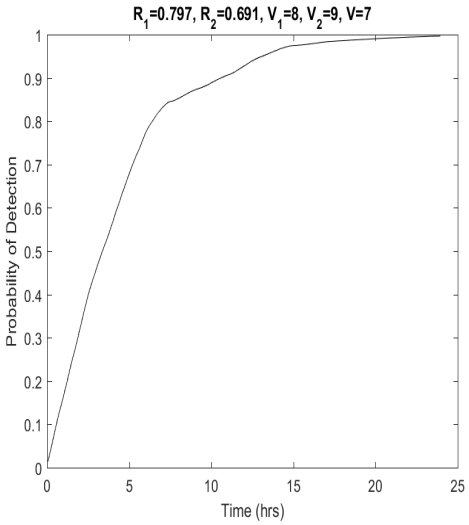


Figure A.74. P_D vs Time

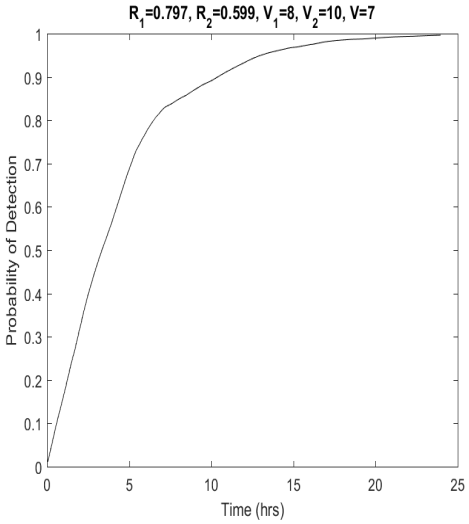


Figure A.75. P_D vs Time

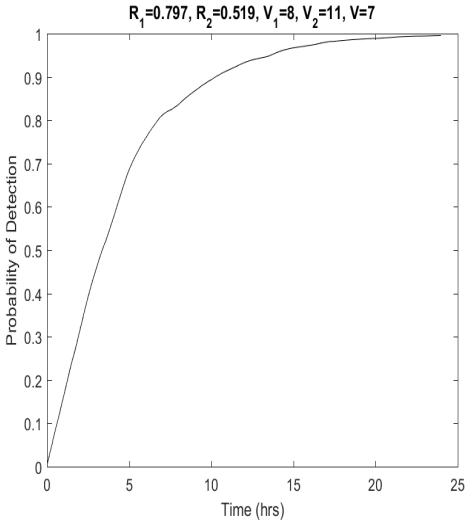


Figure A.76. P_D vs Time

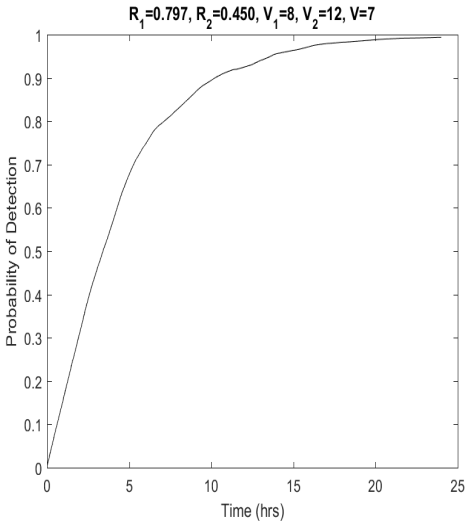


Figure A.77. P_D vs Time

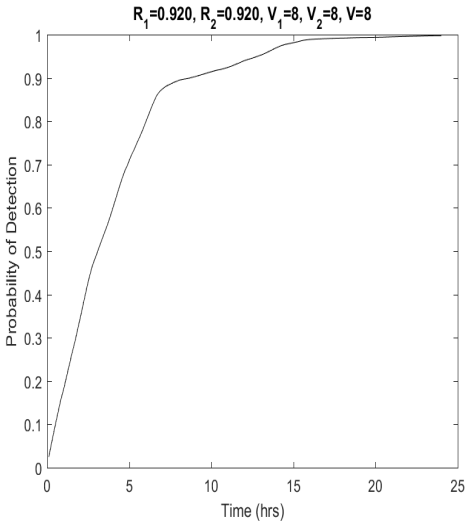


Figure A.78. P_D vs Time

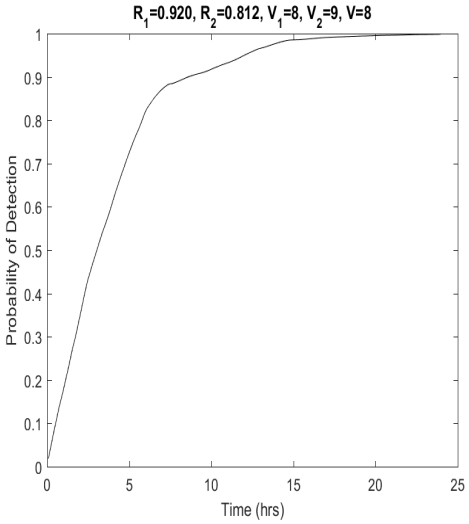


Figure A.79. P_D vs Time

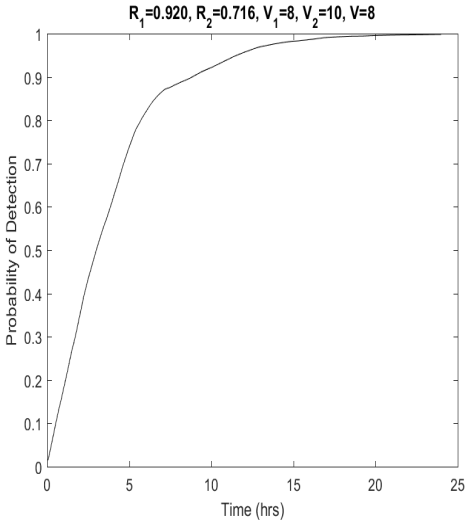


Figure A.80. P_D vs Time

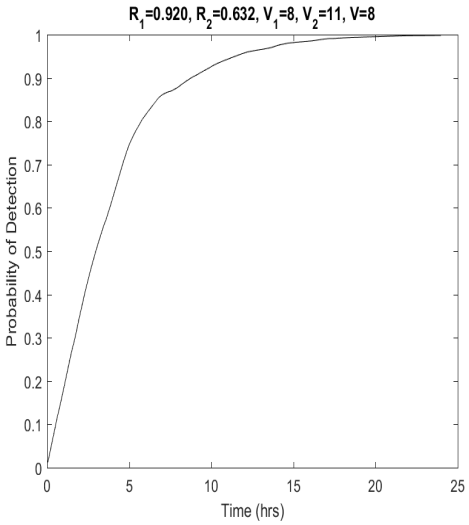


Figure A.81. P_D vs Time

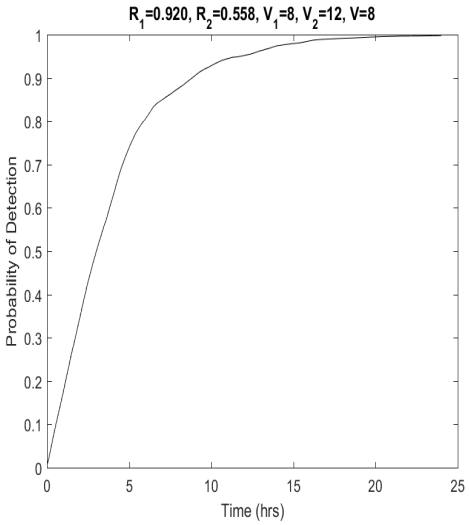


Figure A.82. P_D vs Time

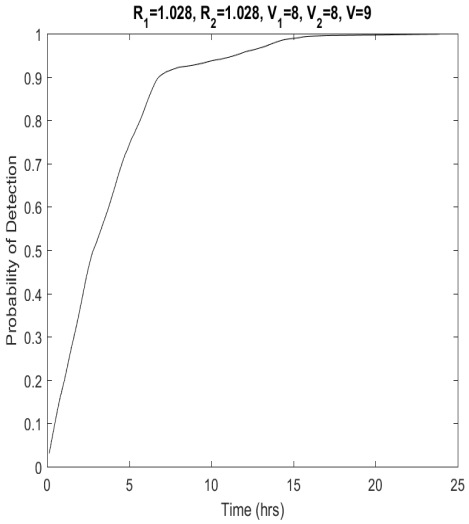


Figure A.83. P_D vs Time

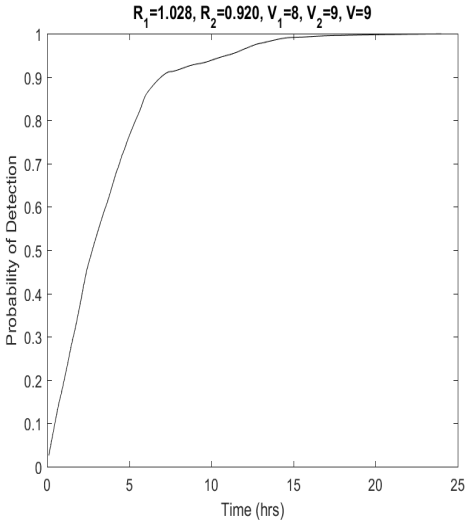


Figure A.84. P_D vs Time

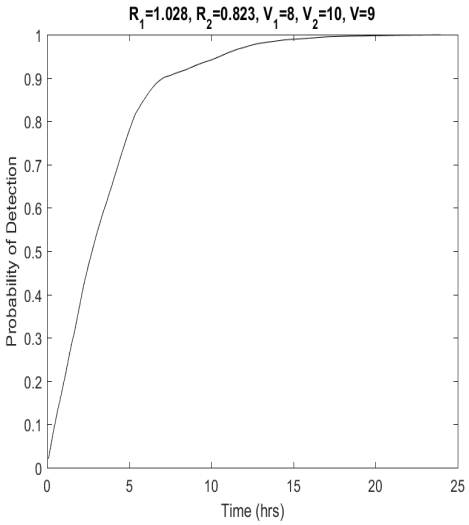


Figure A.85. P_D vs Time

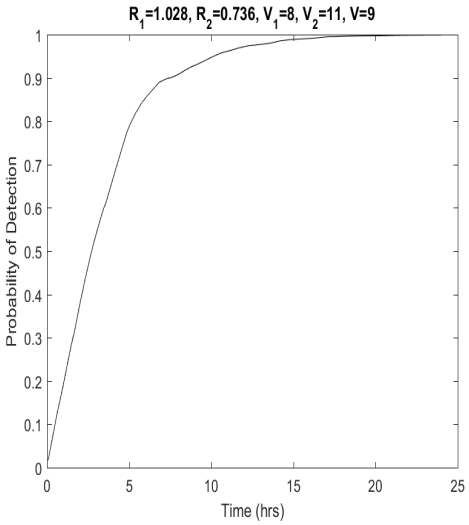


Figure A.86. P_D vs Time

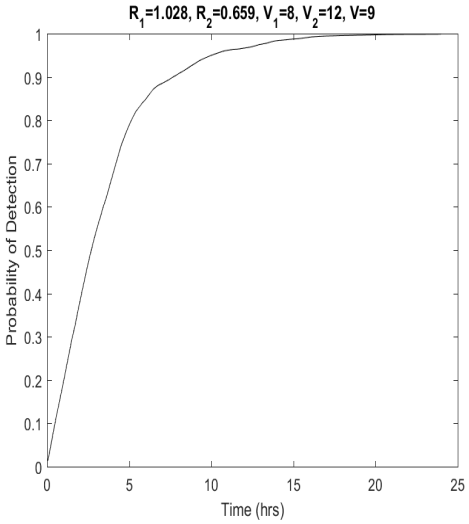


Figure A.87. P_D vs Time

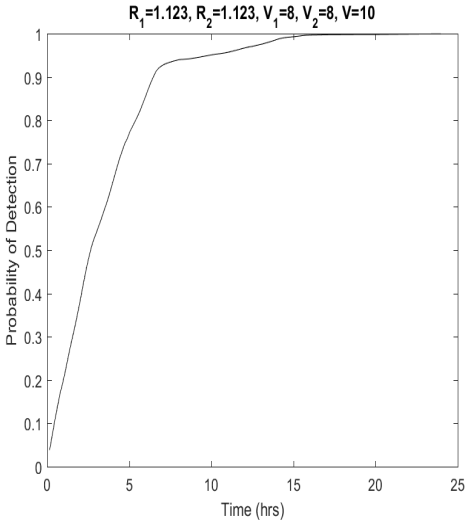


Figure A.88. P_D vs Time

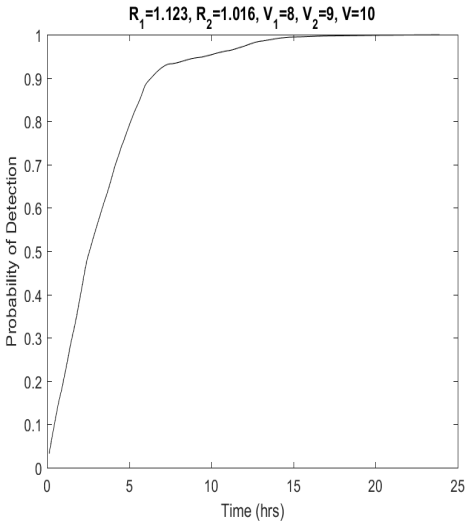


Figure A.89. P_D vs Time

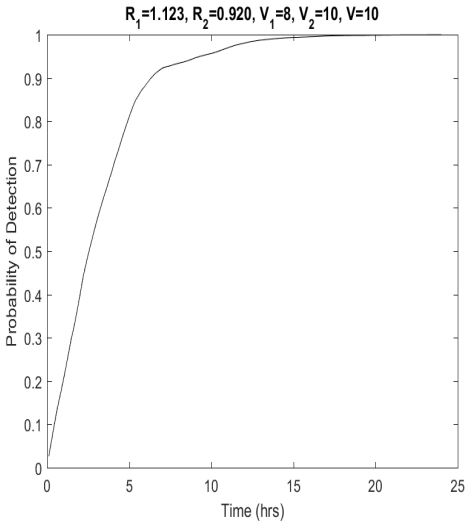


Figure A.90. P_D vs Time

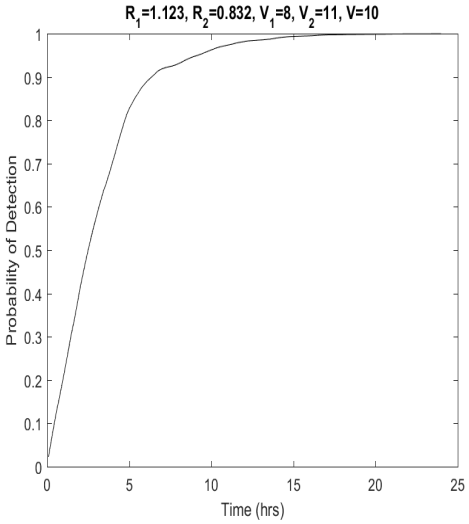


Figure A.91. P_D vs Time

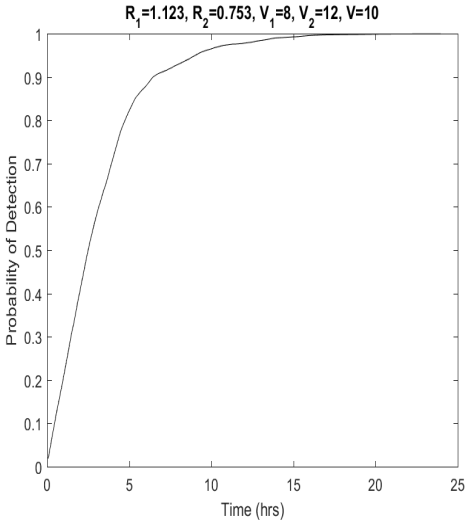


Figure A.92. P_D vs Time

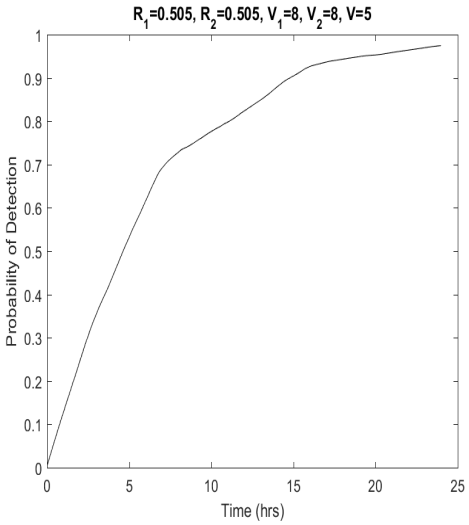


Figure A.93. P_D vs Time

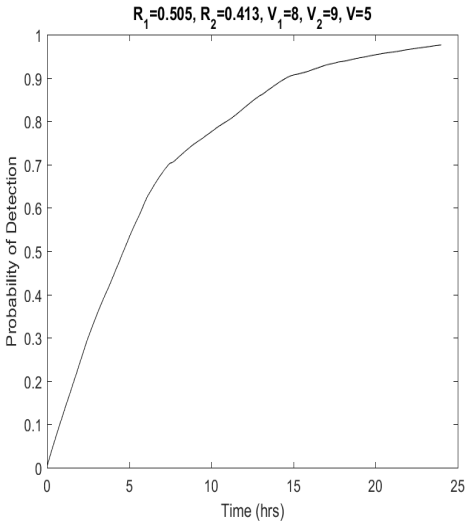


Figure A.94. P_D vs Time

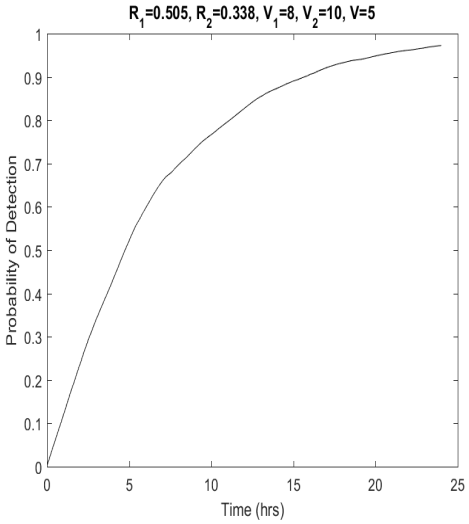


Figure A.95. P_D vs Time

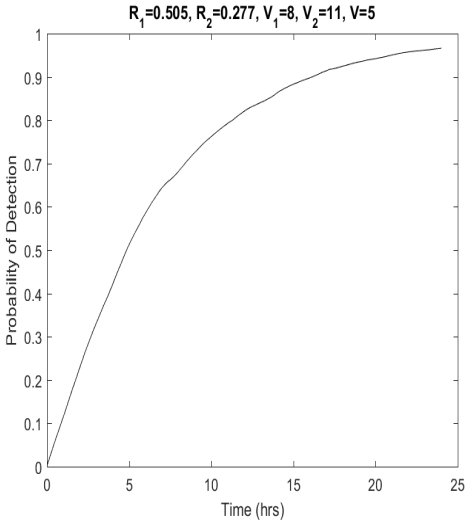


Figure A.96. P_D vs Time

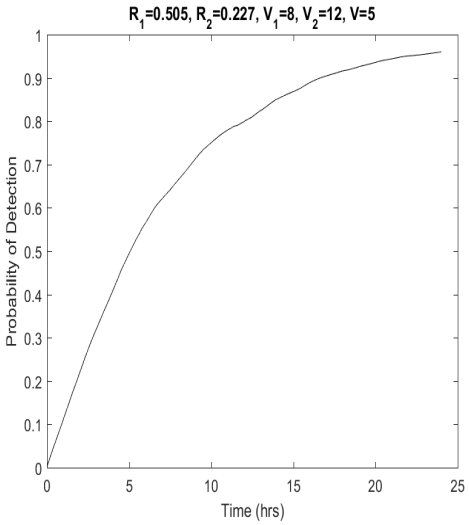


Figure A.97. P_D vs Time

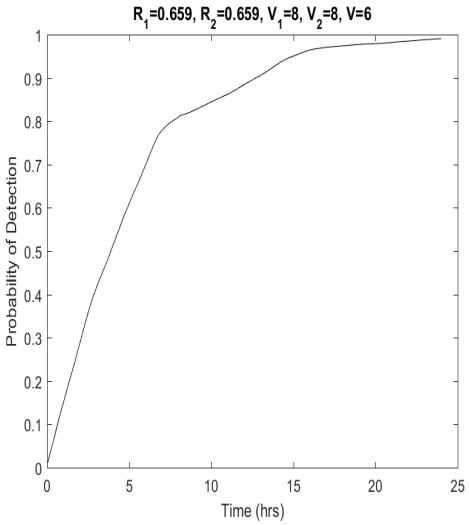


Figure A.98. P_D vs Time

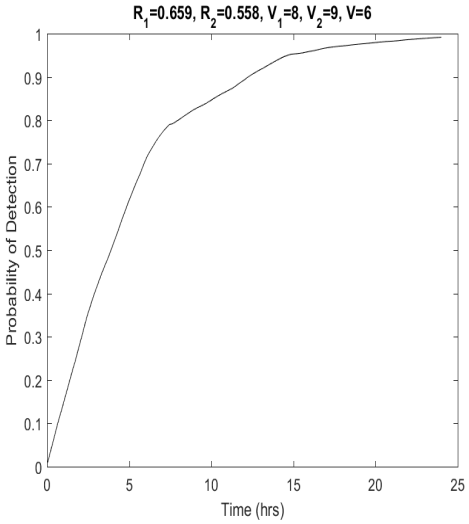


Figure A.99. P_D vs Time

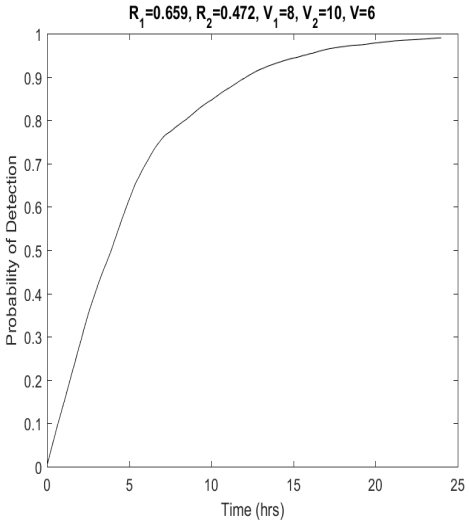


Figure A.100. P_D vs Time

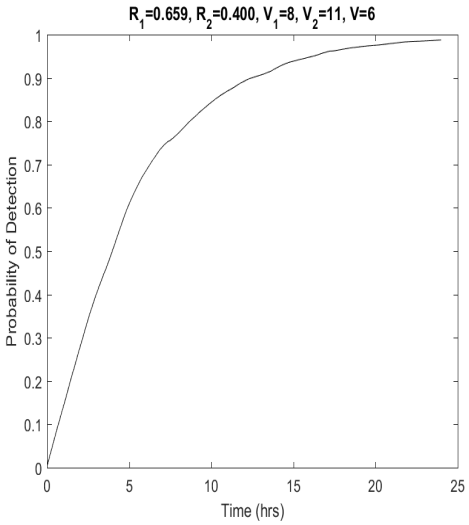


Figure A.101. P_D vs Time

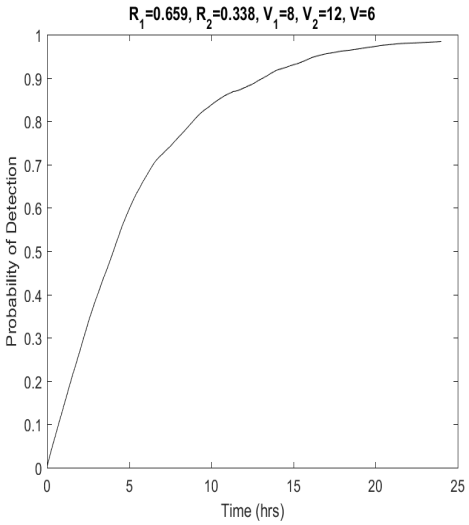


Figure A.102. P_D vs Time

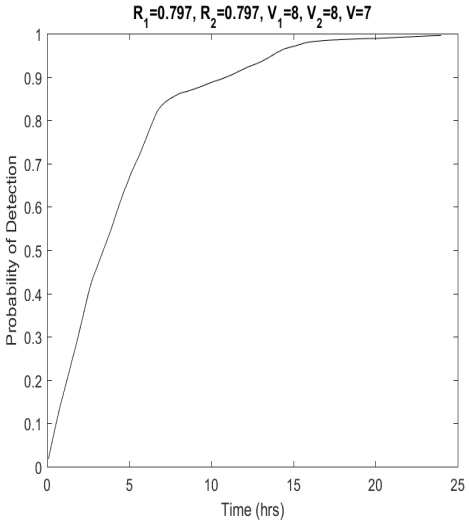


Figure A.103. P_D vs Time

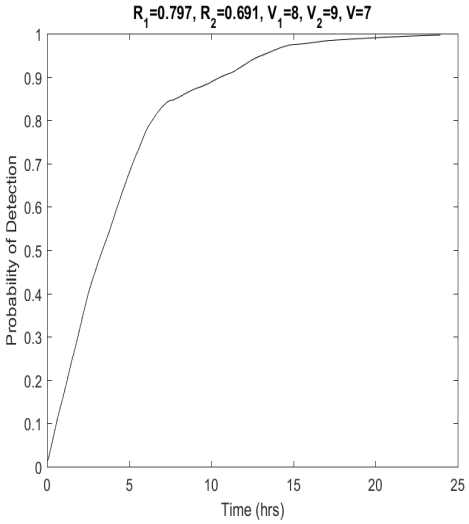


Figure A.104. P_D vs Time

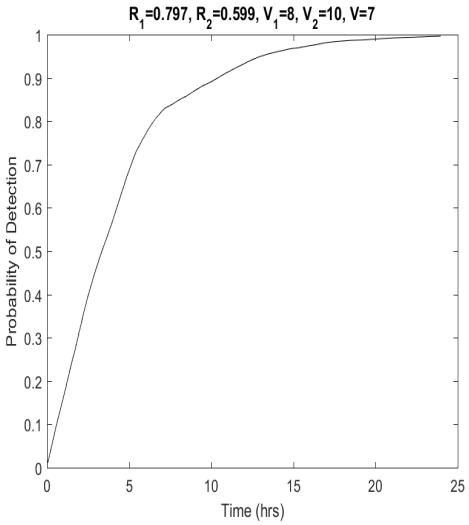


Figure A.105. P_D vs Time

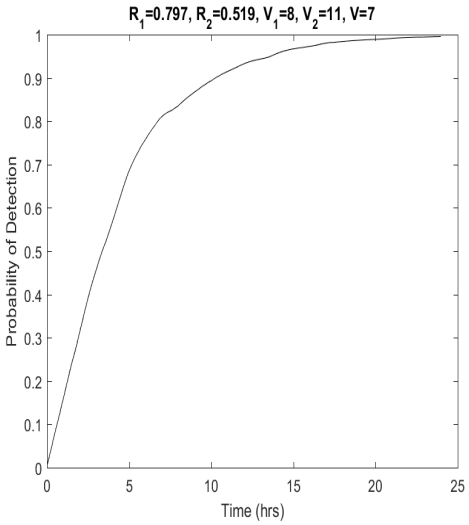


Figure A.106. P_D vs Time

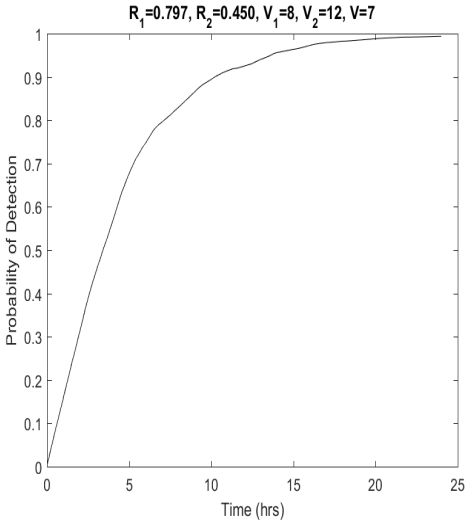


Figure A.107. P_D vs Time

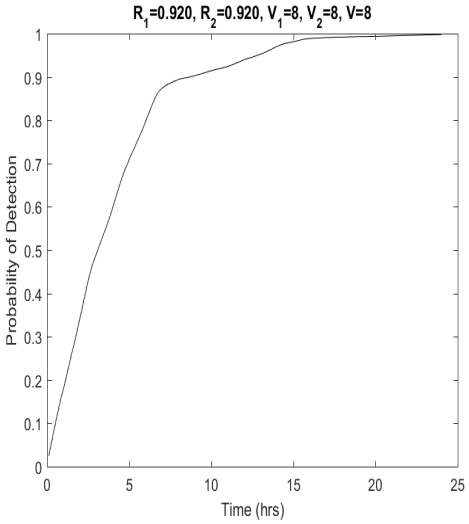


Figure A.108. P_D vs Time

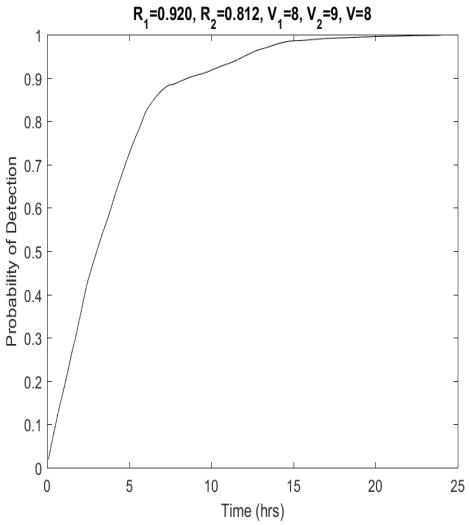


Figure A.109. P_D vs Time

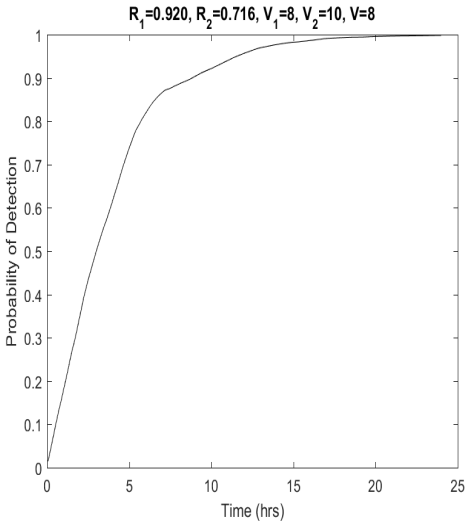


Figure A.110. P_D vs Time

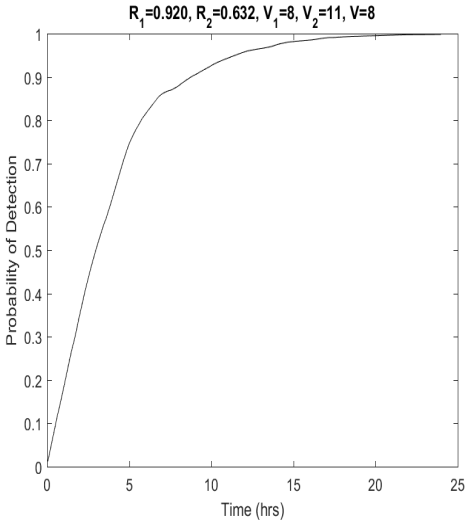


Figure A.111. P_D vs Time

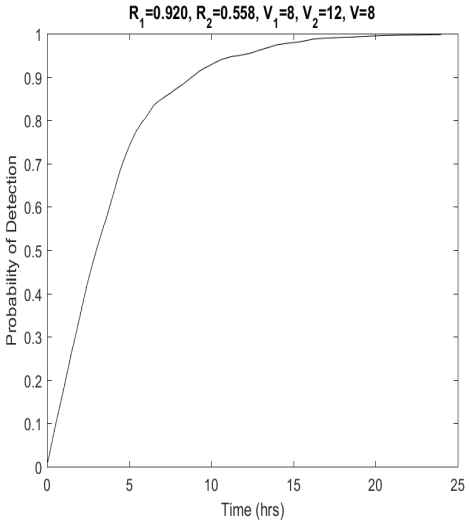


Figure A.112. P_D vs Time

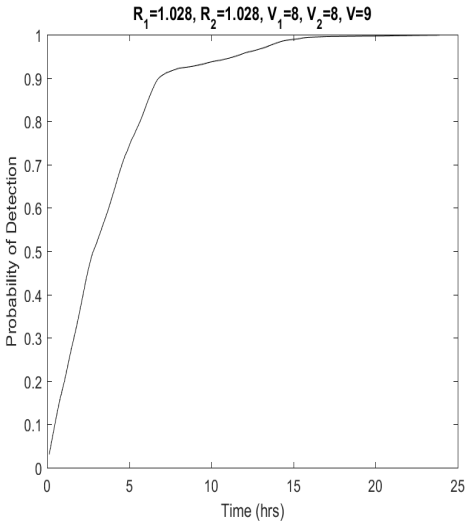


Figure A.113. P_D vs Time

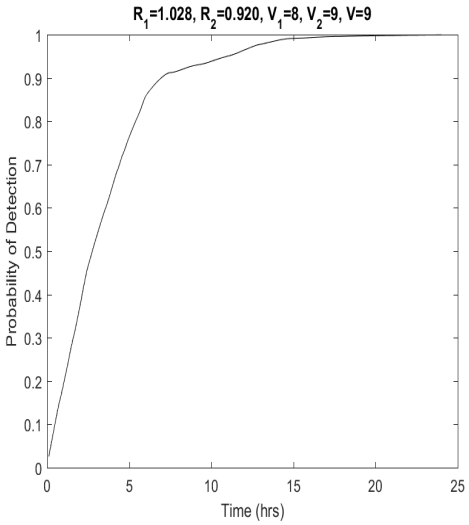


Figure A.114. P_D vs Time

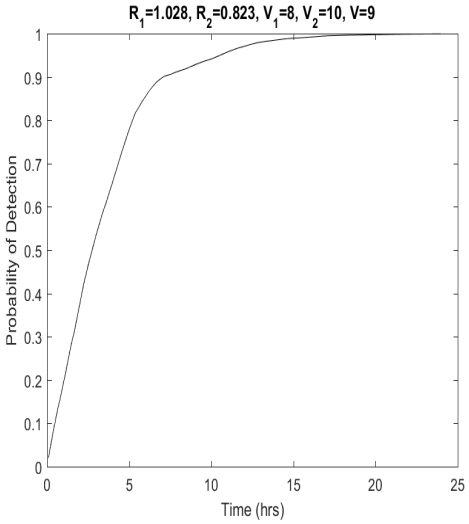


Figure A.115. P_D vs Time

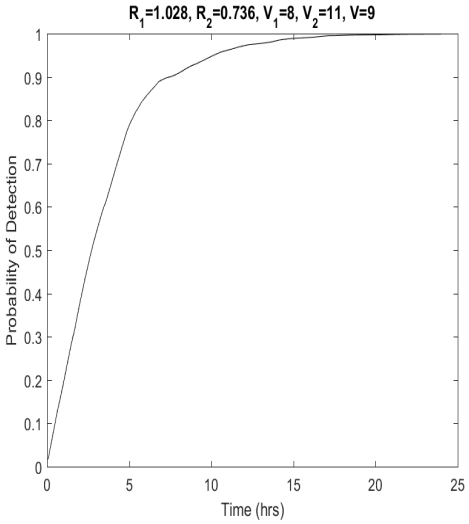


Figure A.116. P_D vs Time

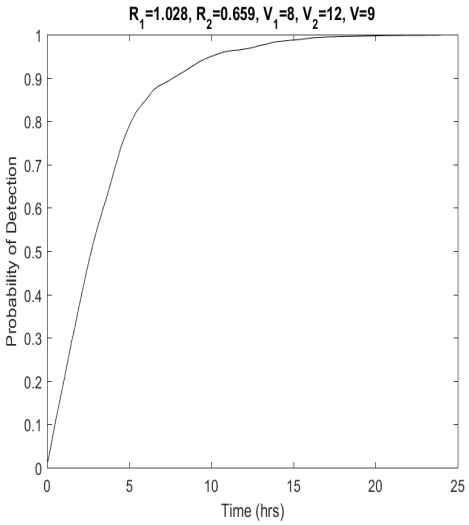


Figure A.117. P_D vs Time

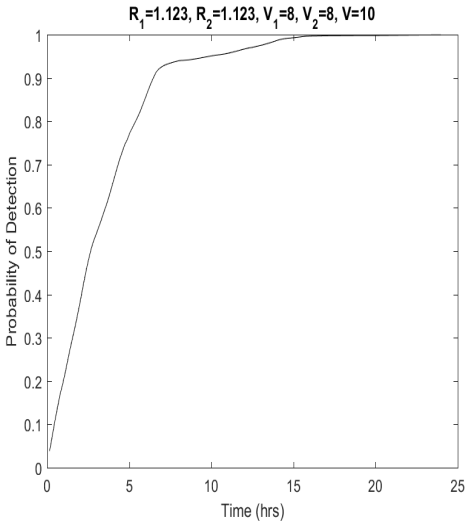


Figure A.118. P_D vs Time

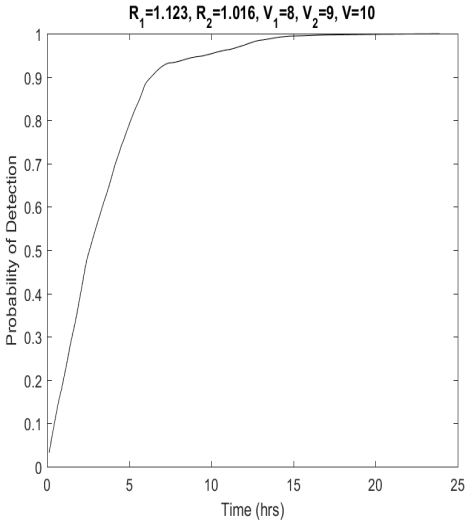


Figure A.119. P_D vs Time

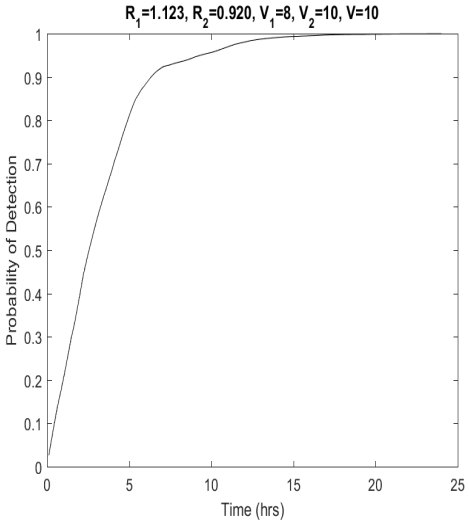


Figure A.120. P_D vs Time

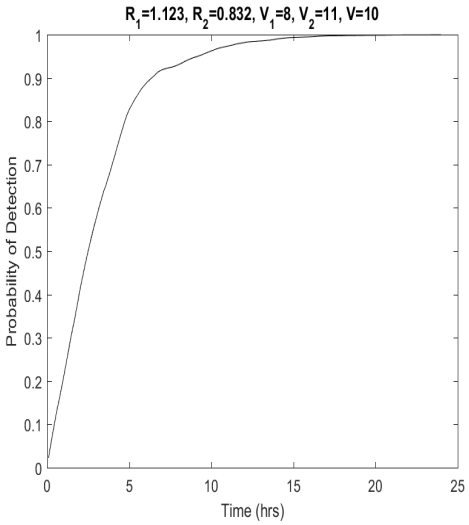


Figure A.121. P_D vs Time

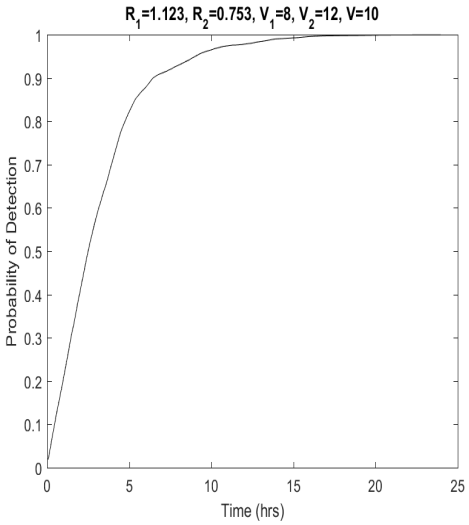


Figure A.122. P_D vs Time

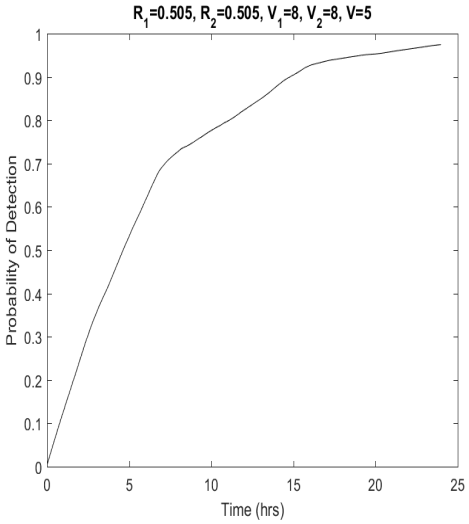


Figure A.123. P_D vs Time

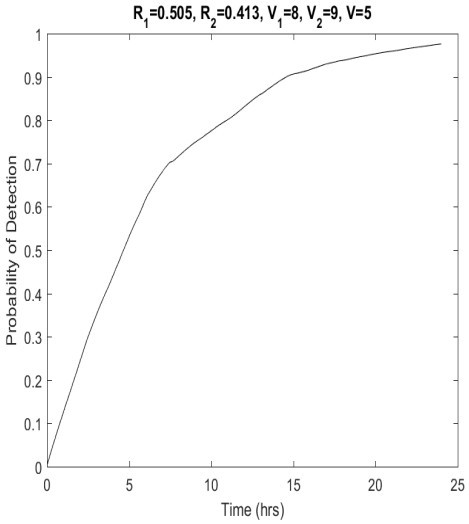


Figure A.124. P_D vs Time

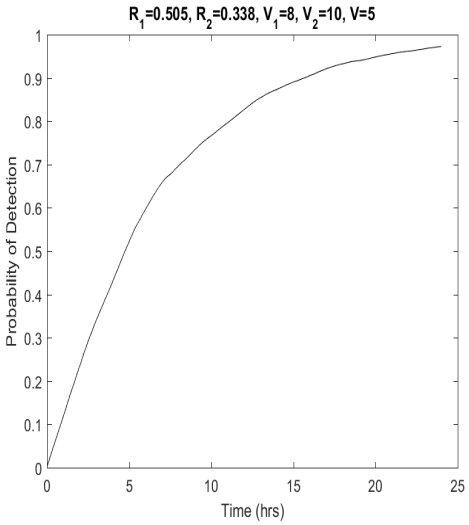


Figure A.125. P_D vs Time

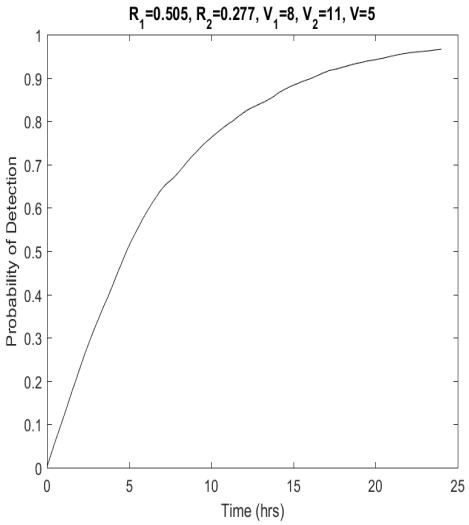


Figure A.126. P_D vs Time

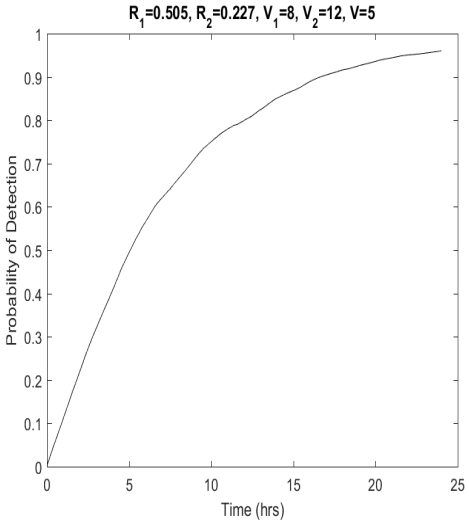


Figure A.127. P_D vs Time

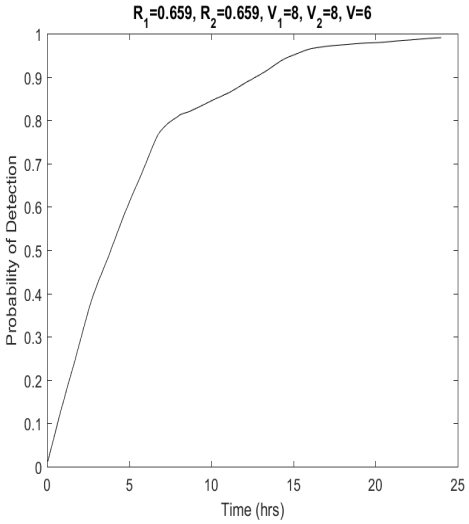


Figure A.128. P_D vs Time

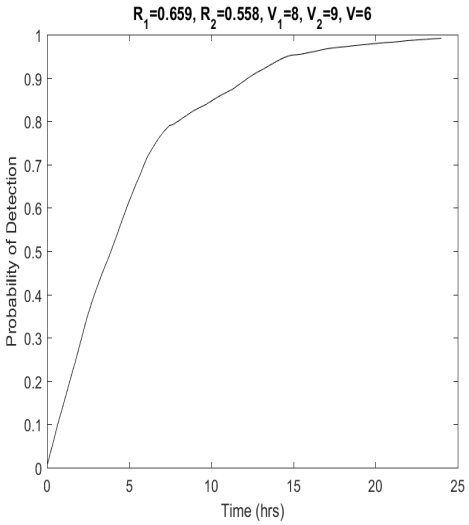


Figure A.129. P_D vs Time

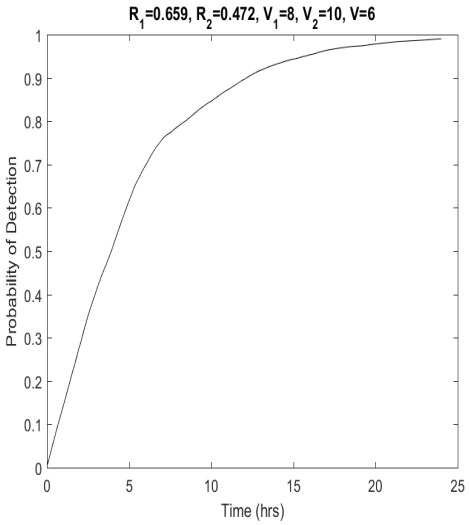


Figure A.130. P_D vs Time

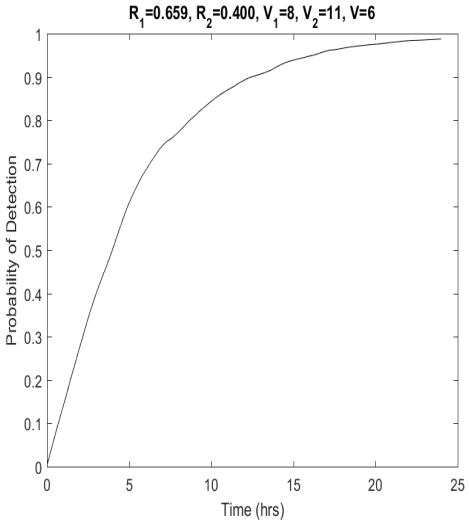


Figure A.131. P_D vs Time

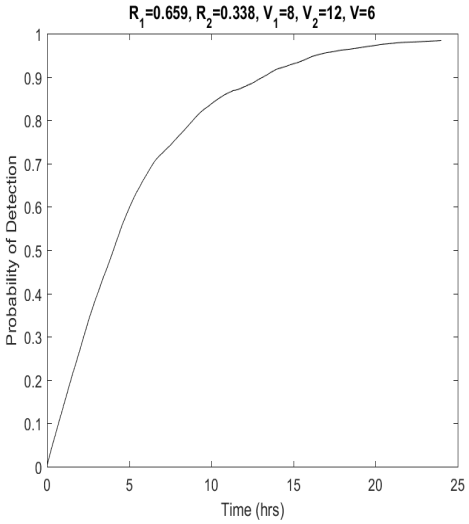


Figure A.132. P_D vs Time

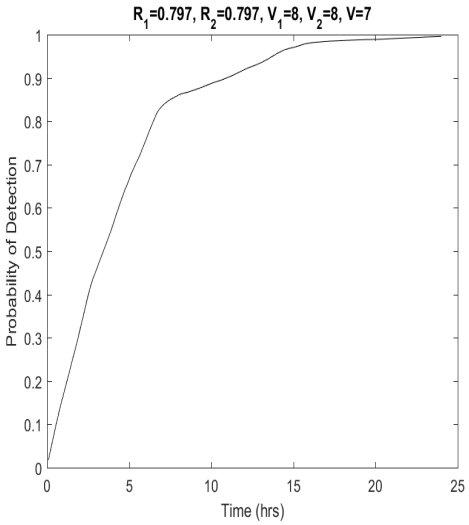


Figure A.133. P_D vs Time

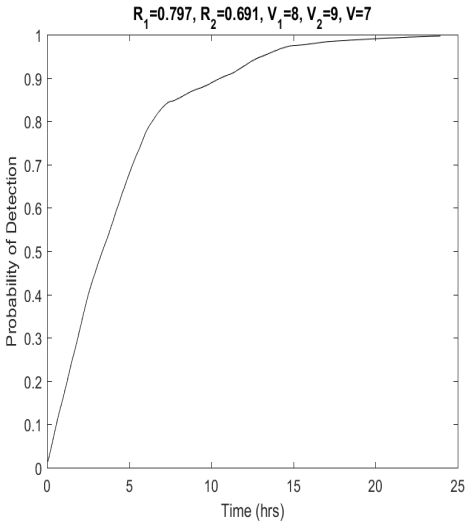


Figure A.134. P_D vs Time

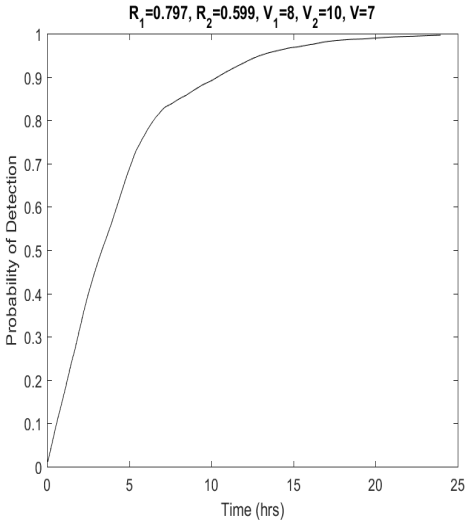


Figure A.135. P_D vs Time

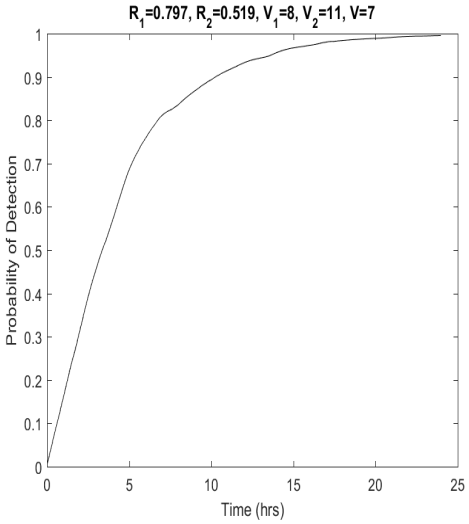


Figure A.136. P_D vs Time

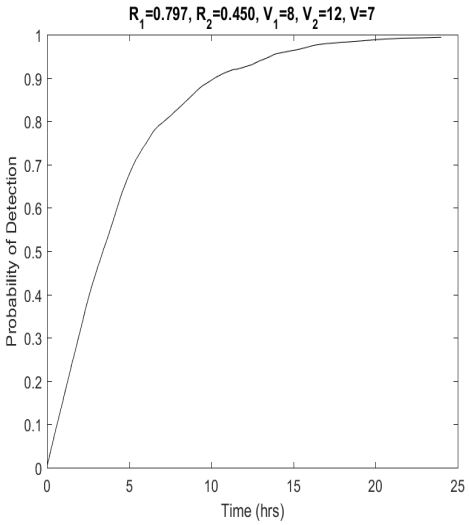


Figure A.137. P_D vs Time

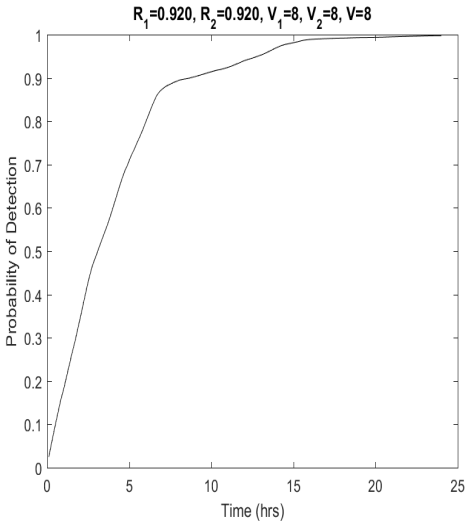


Figure A.138. P_D vs Time

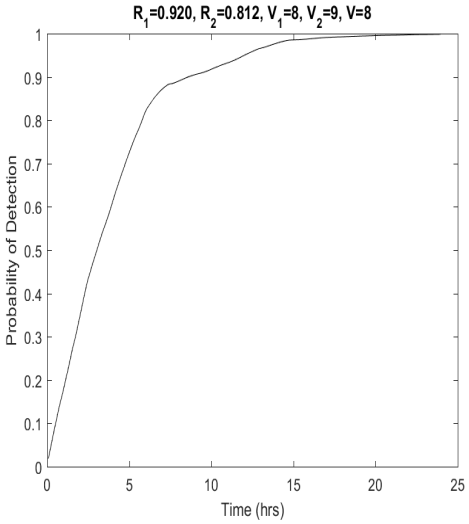


Figure A.139. P_D vs Time

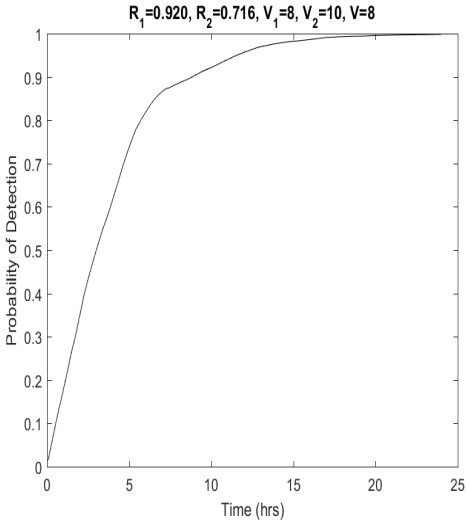


Figure A.140. P_D vs Time

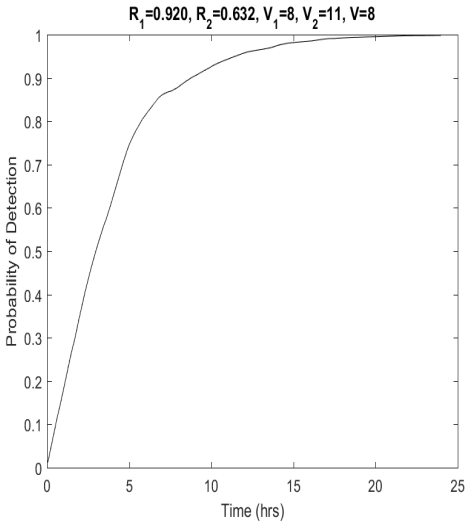


Figure A.141. P_D vs Time

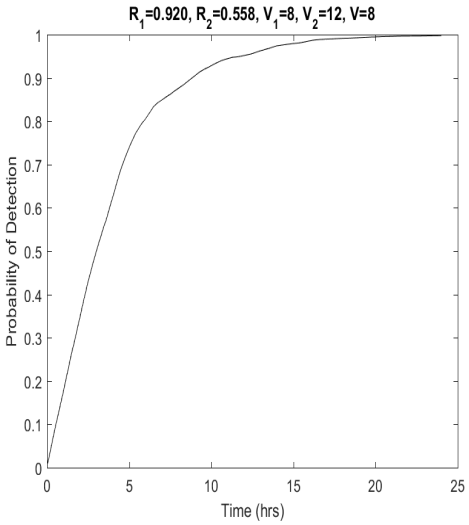


Figure A.142. P_D vs Time

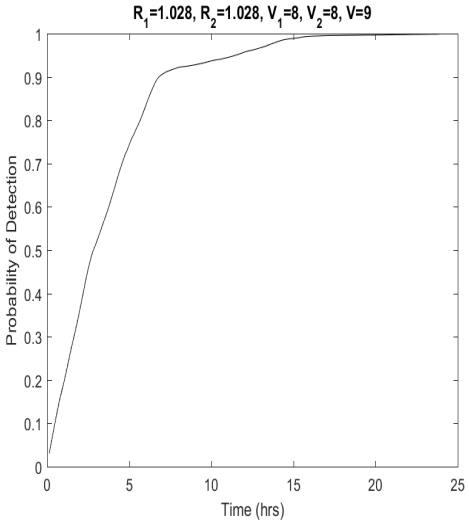


Figure A.143. P_D vs Time

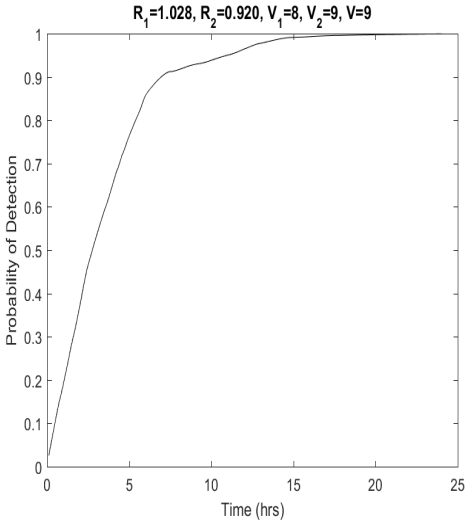


Figure A.144. P_D vs Time

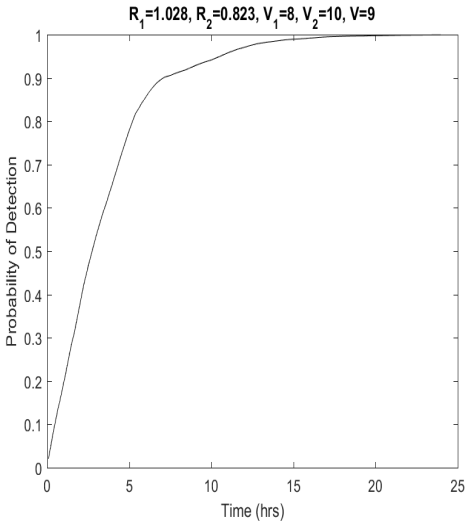


Figure A.145. P_D vs Time

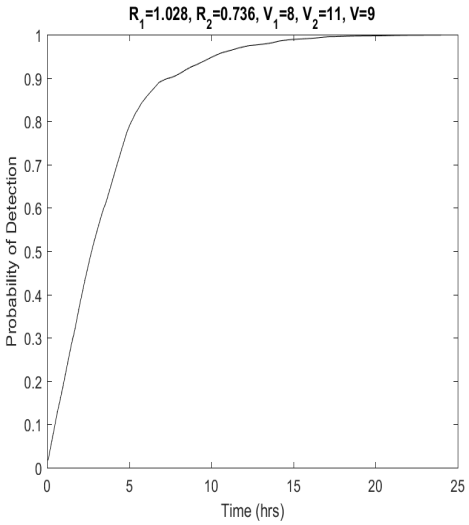


Figure A.146. P_D vs Time

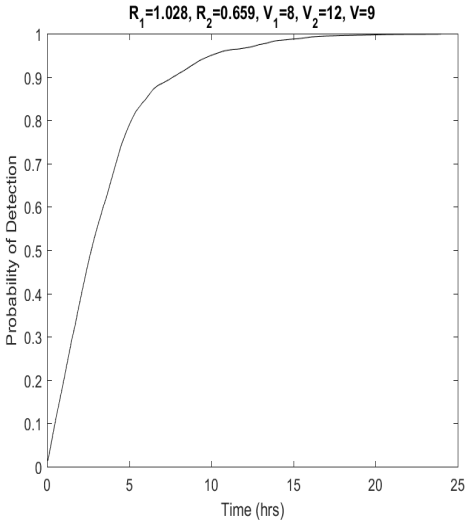


Figure A.147. P_D vs Time

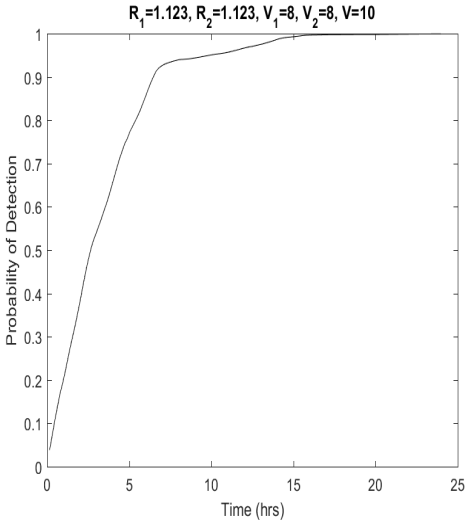


Figure A.148. P_D vs Time

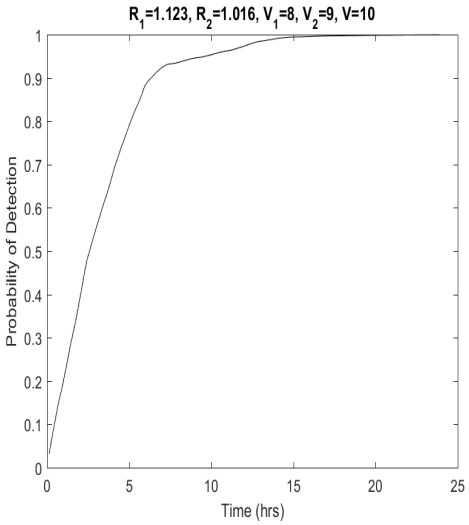


Figure A.149. P_D vs Time

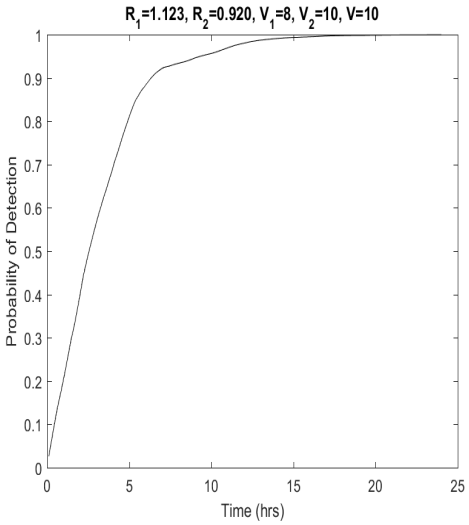


Figure A.150. P_D vs Time

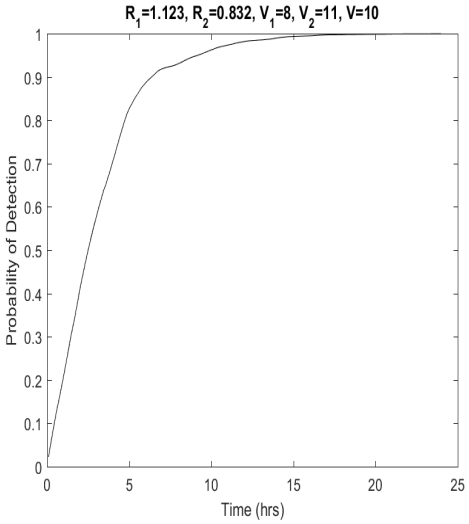


Figure A.151. P_D vs Time

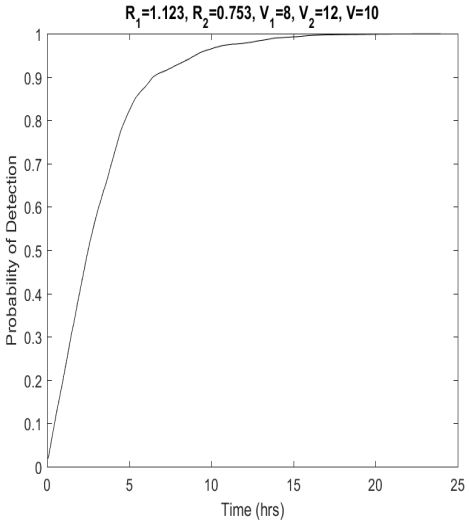


Figure A.152. P_D vs Time

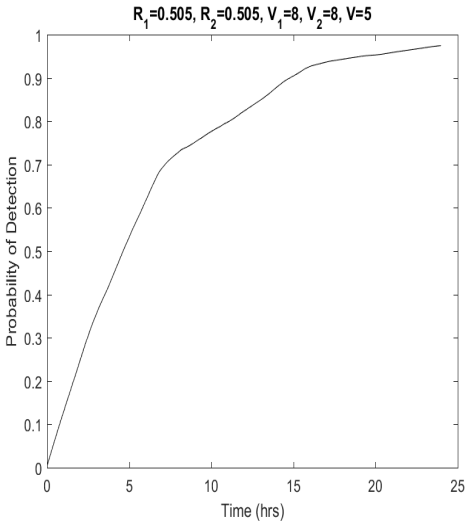


Figure A.153. P_D vs Time

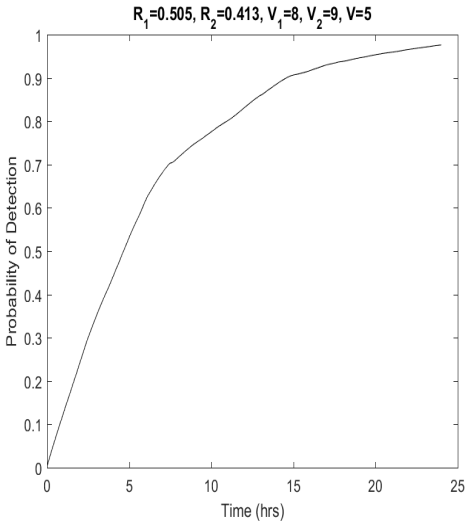


Figure A.154. P_D vs Time

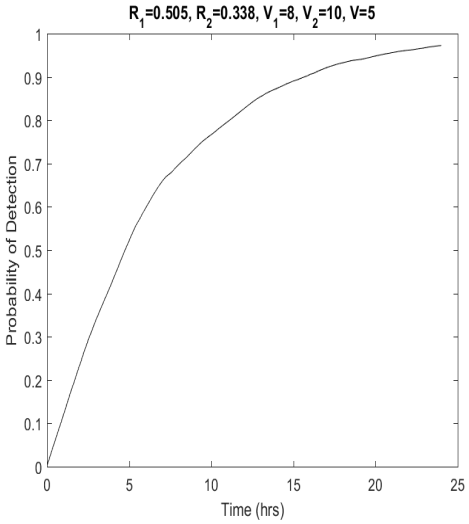


Figure A.155. P_D vs Time

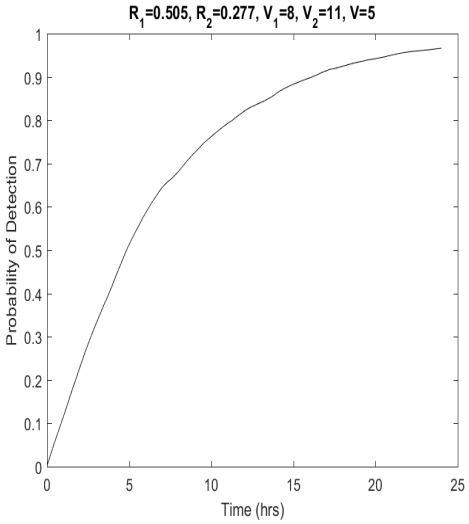


Figure A.156. P_D vs Time

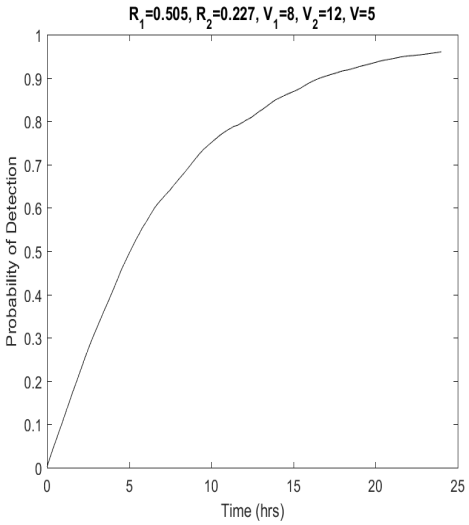


Figure A.157. P_D vs Time

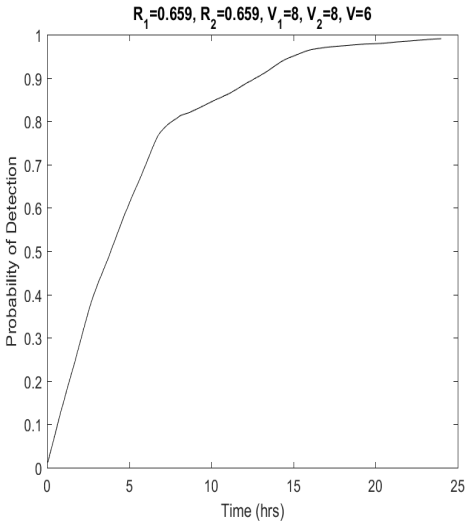


Figure A.158. P_D vs Time

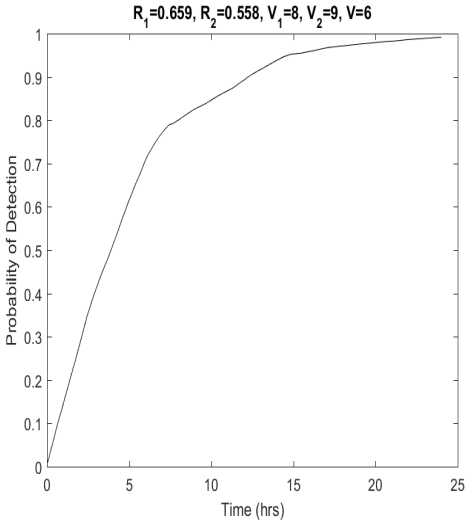


Figure A.159. P_D vs Time

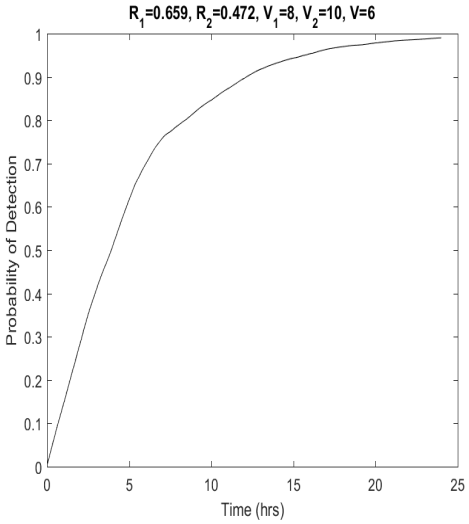


Figure A.160. P_D vs Time

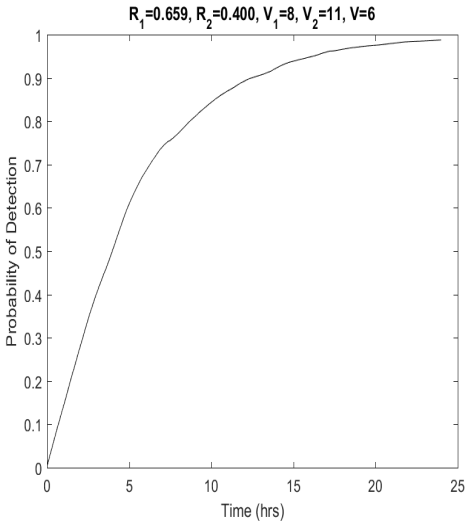


Figure A.161. P_D vs Time

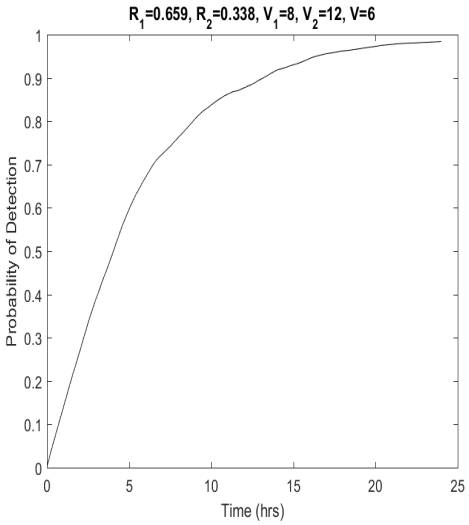


Figure A.162. P_D vs Time

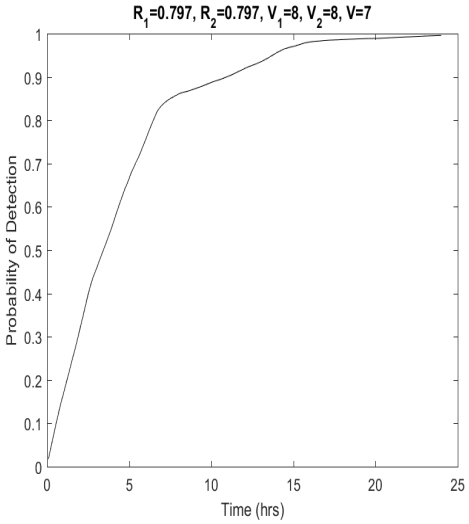


Figure A.163. P_D vs Time

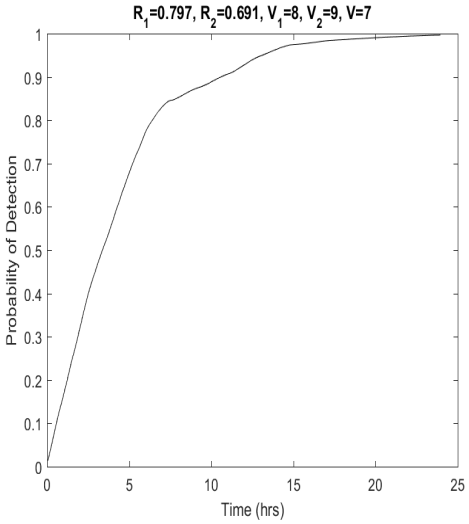


Figure A.164. P_D vs Time

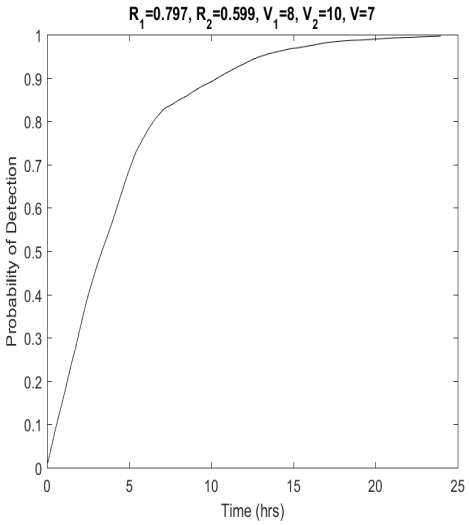


Figure A.165. P_D vs Time

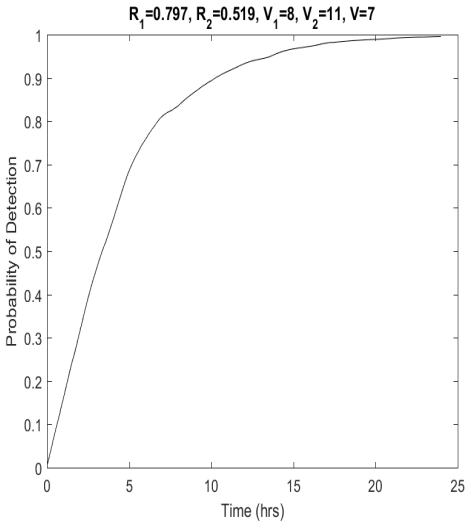


Figure A.166. P_D vs Time

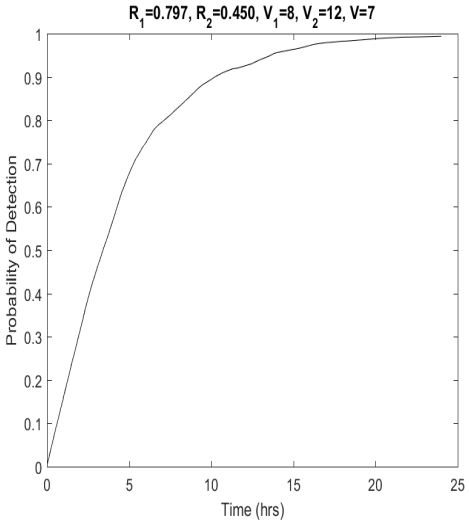


Figure A.167. P_D vs Time

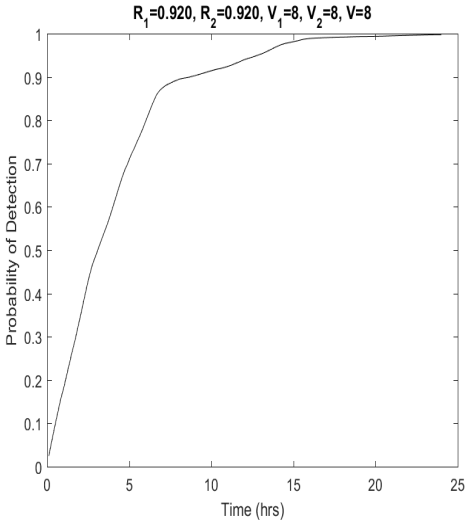


Figure A.168. P_D vs Time

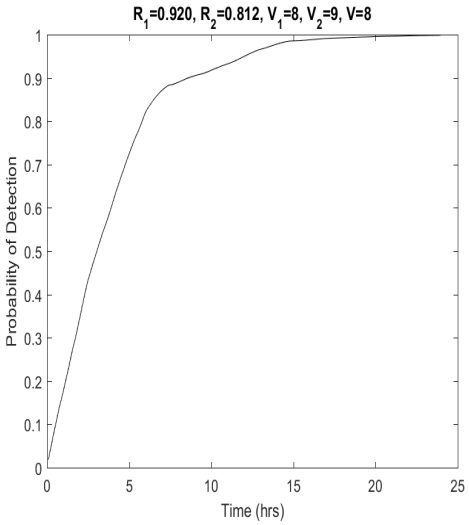


Figure A.169. P_D vs Time

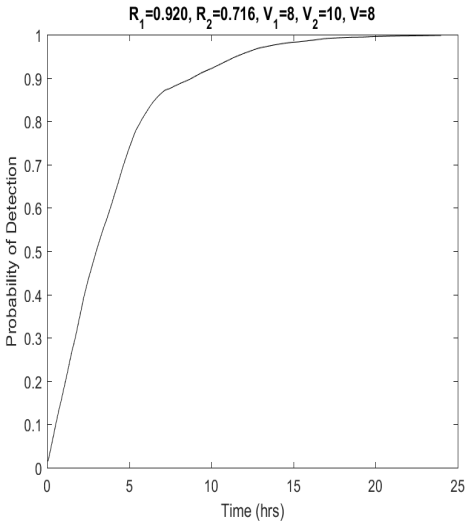


Figure A.170. P_D vs Time

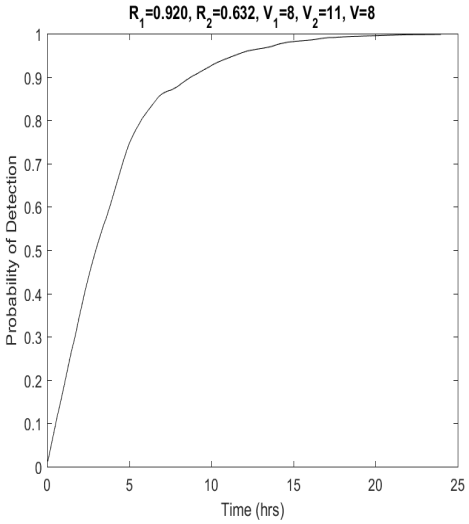


Figure A.171. P_D vs Time

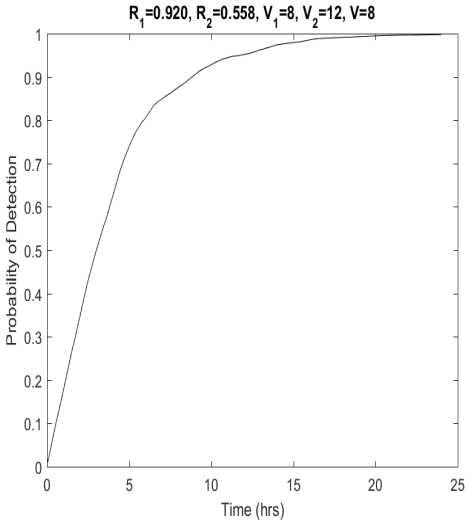


Figure A.172. P_D vs Time

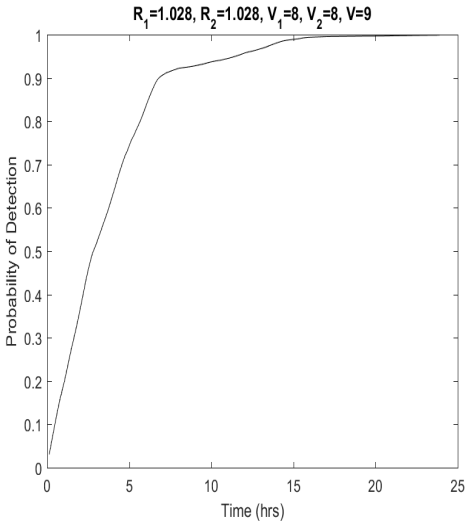


Figure A.173. P_D vs Time

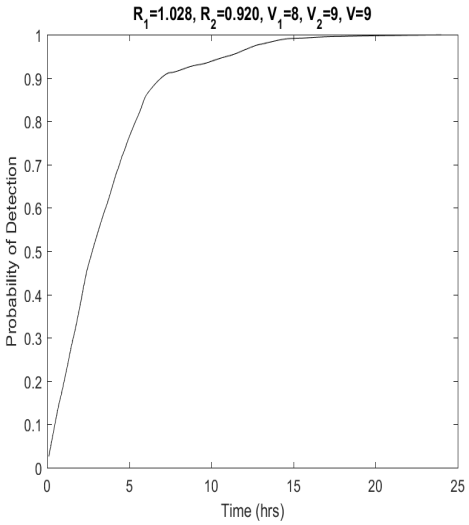


Figure A.174. P_D vs Time

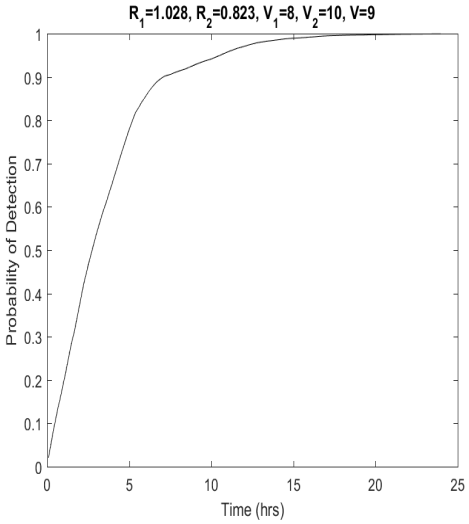


Figure A.175. P_D vs Time

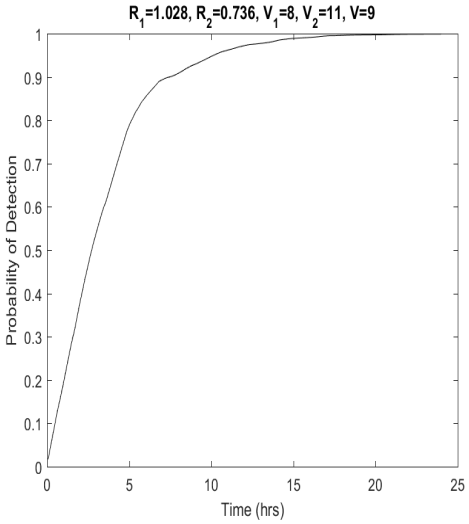


Figure A.176. P_D vs Time

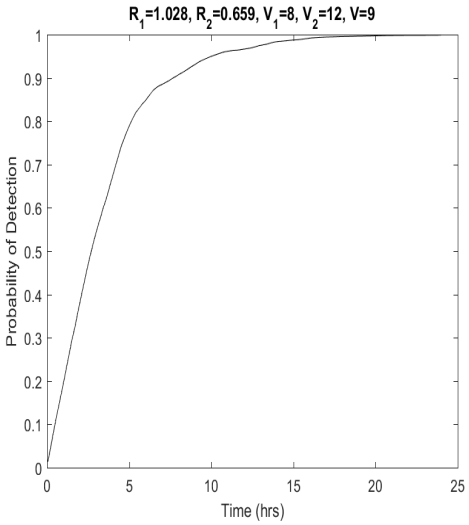


Figure A.177. P_D vs Time

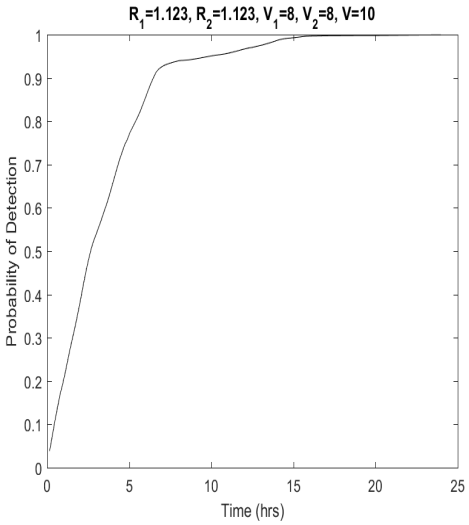


Figure A.178. P_D vs Time

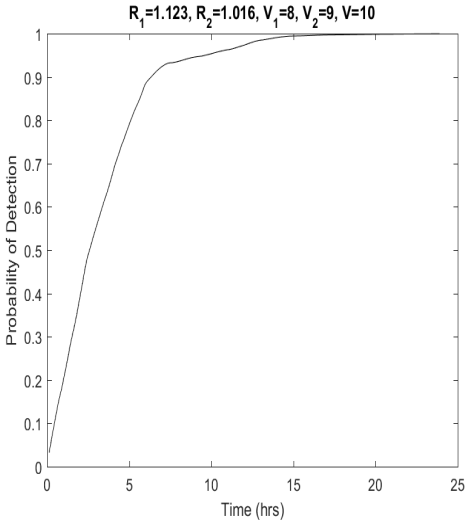


Figure A.179. P_D vs Time

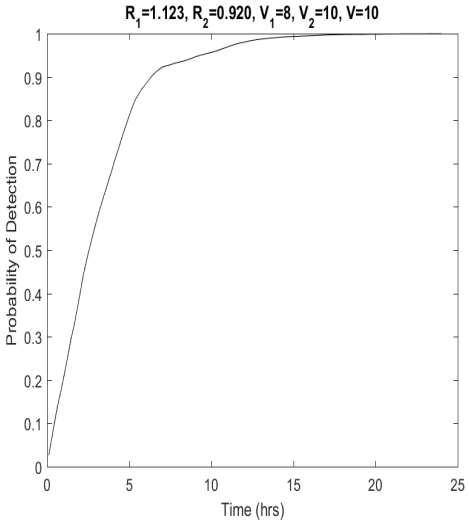


Figure A.180. P_D vs Time

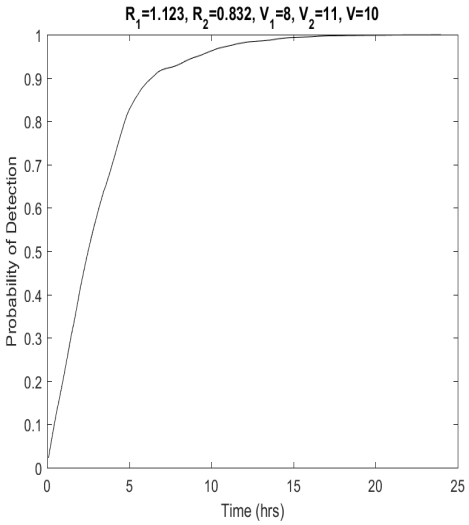


Figure A.181. P_D vs Time

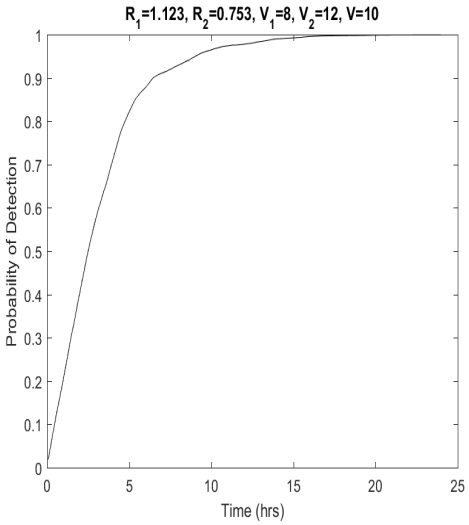


Figure A.182. P_D vs Time

A.3.2 Multipath Ladder Search

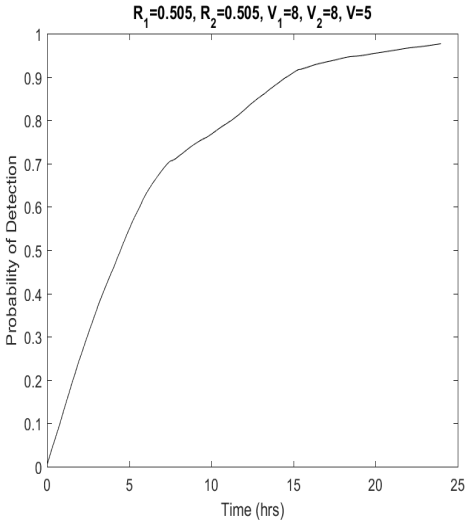


Figure A.183. P_D vs Time

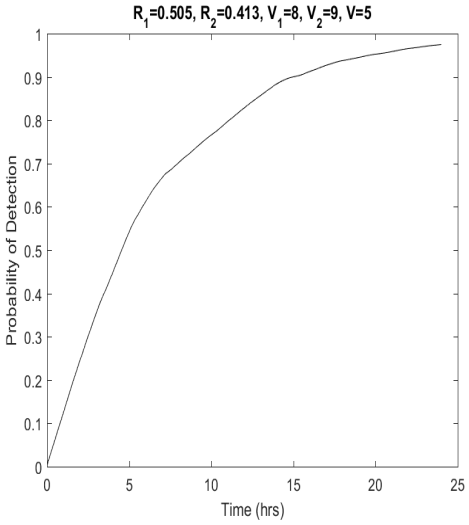


Figure A.184. P_D vs Time

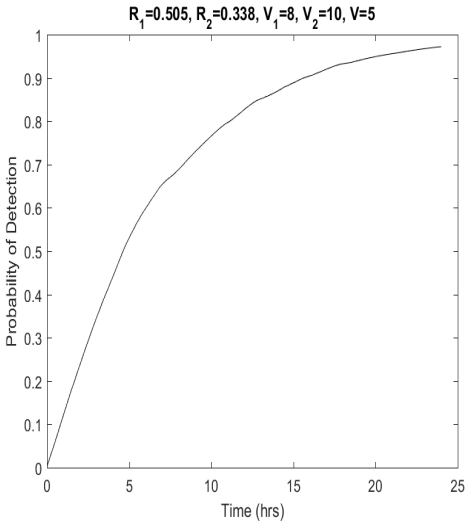


Figure A.185. P_D vs Time

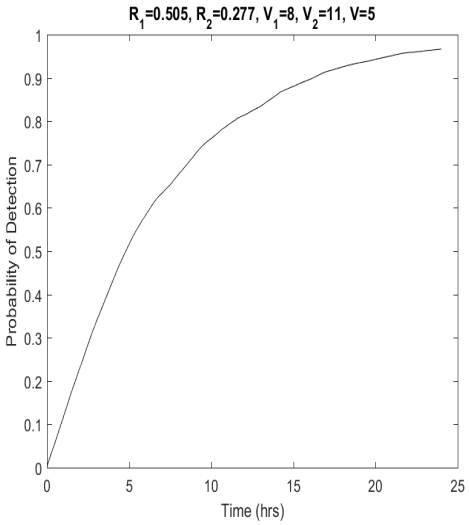


Figure A.186. P_D vs Time

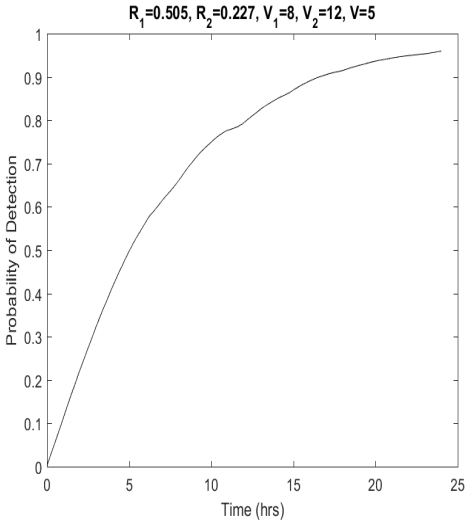


Figure A.187. P_D vs Time

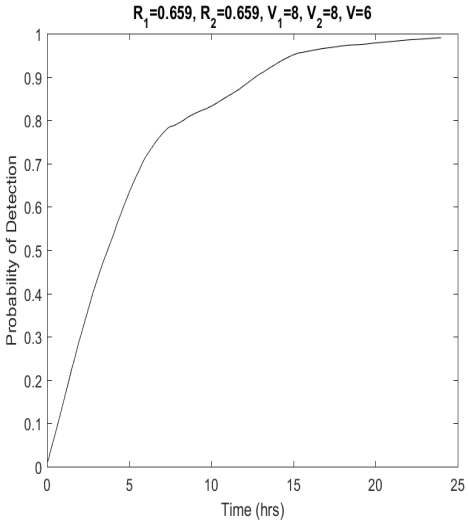


Figure A.188. P_D vs Time

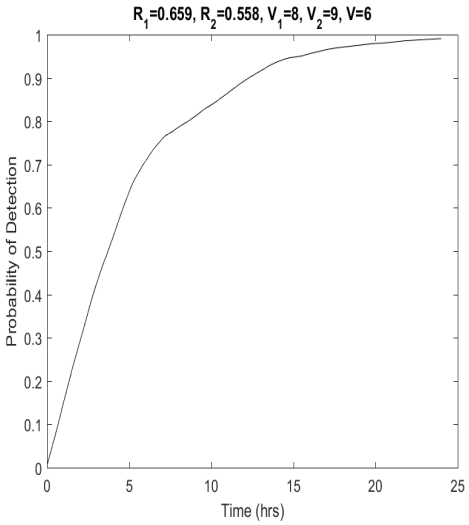


Figure A.189. P_D vs Time

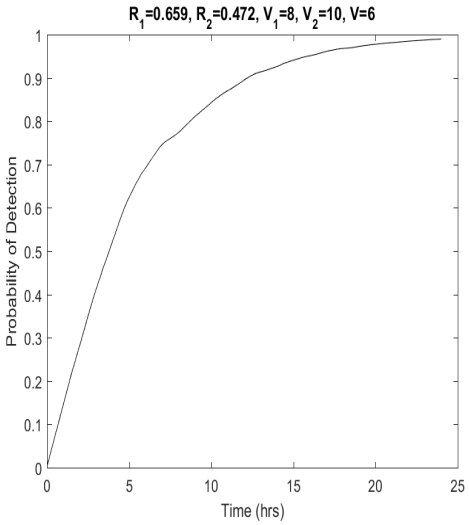


Figure A.190. P_D vs Time

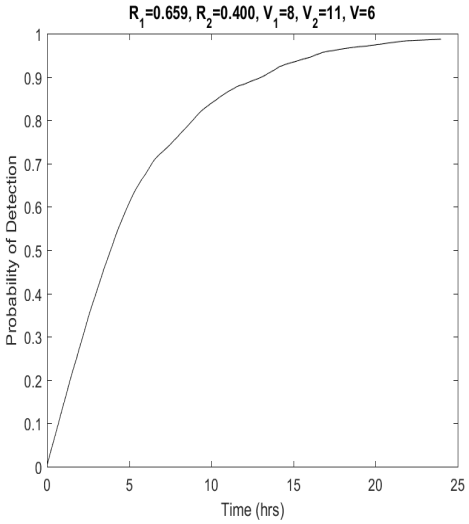


Figure A.191. P_D vs Time

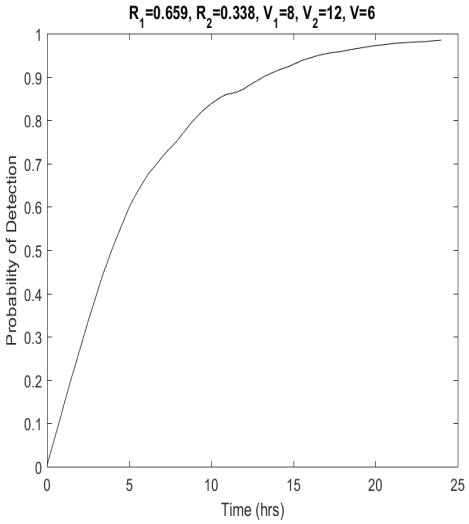


Figure A.192. P_D vs Time

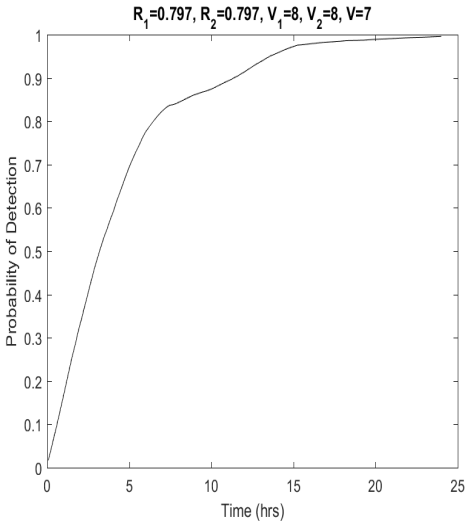


Figure A.193. P_D vs Time

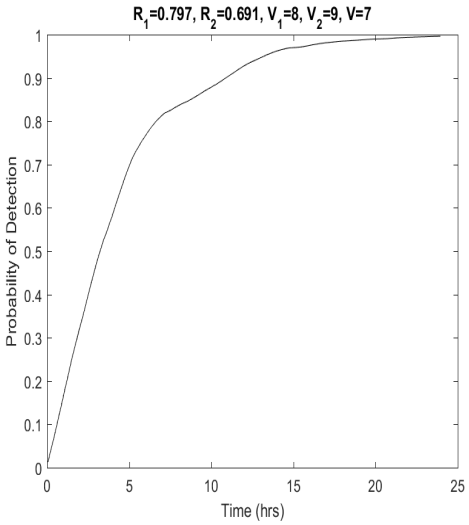


Figure A.194. P_D vs Time

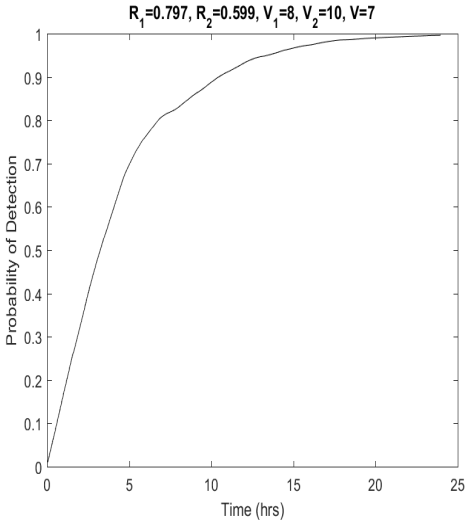


Figure A.195. P_D vs Time

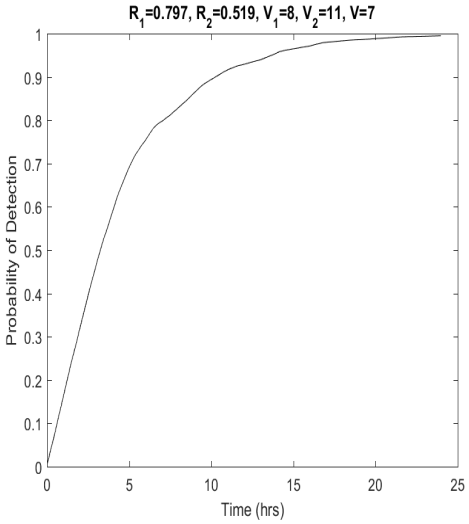


Figure A.196. P_D vs Time

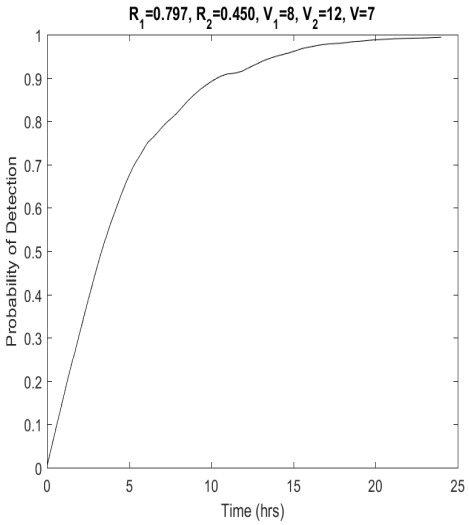


Figure A.197. P_D vs Time

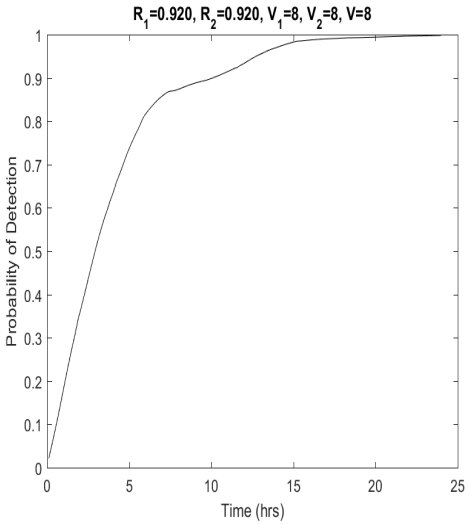


Figure A.198. P_D vs Time

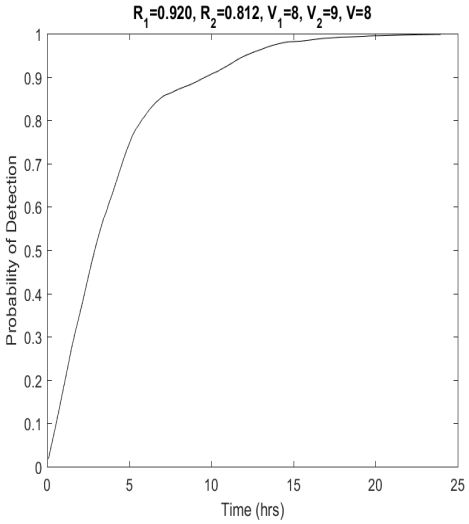


Figure A.199. P_D vs Time

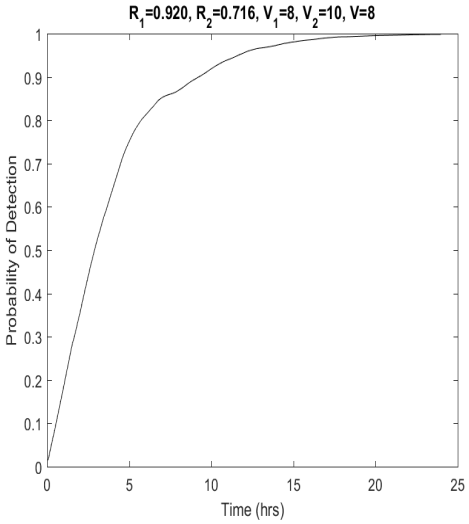


Figure A.200. P_D vs Time

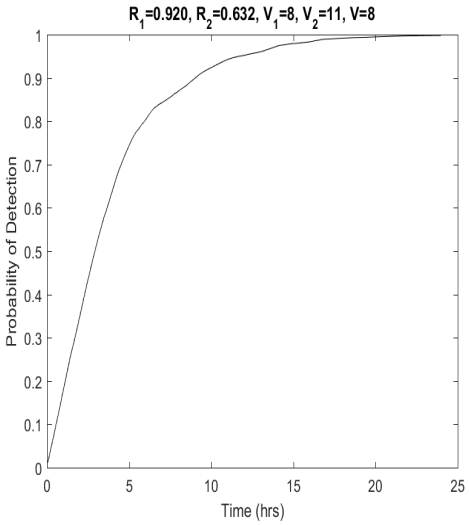


Figure A.201. P_D vs Time

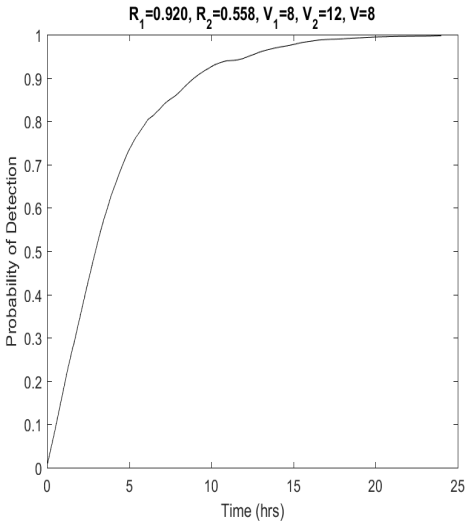


Figure A.202. P_D vs Time

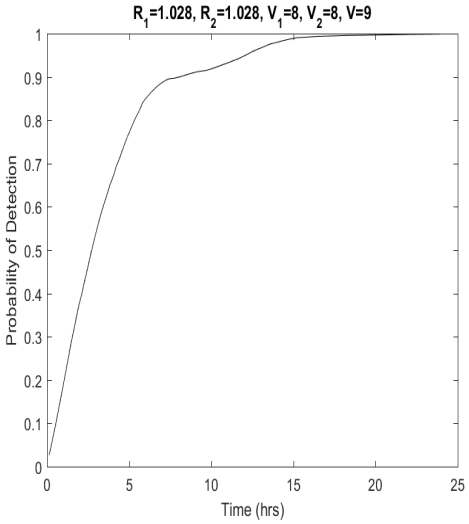


Figure A.203. P_D vs Time

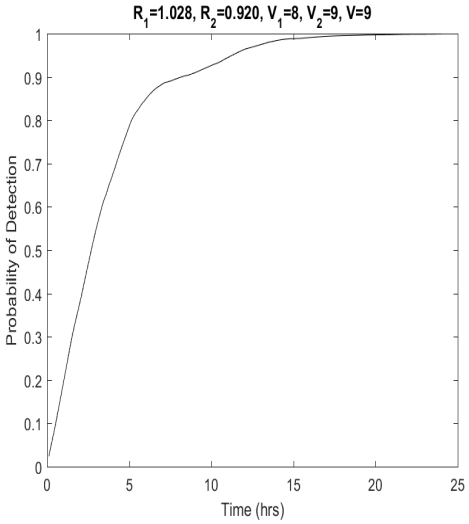


Figure A.204. P_D vs Time

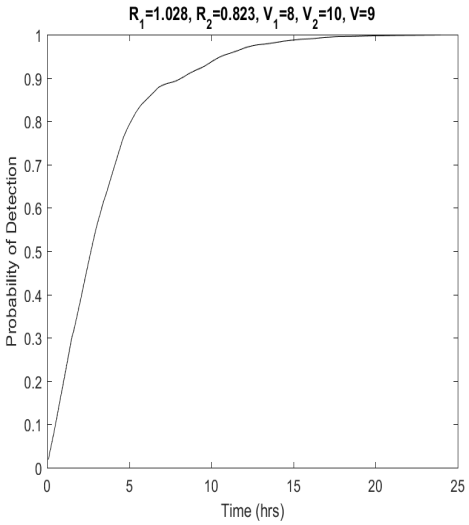


Figure A.205. P_D vs Time

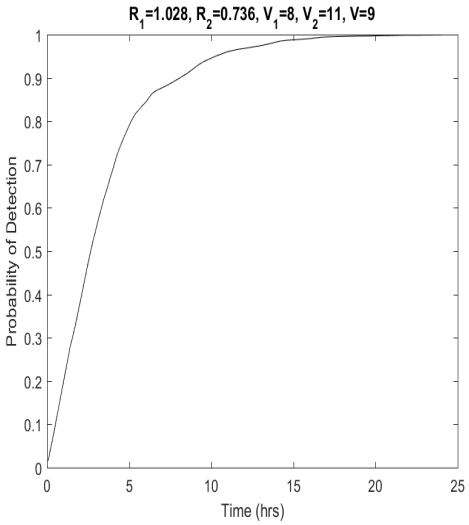


Figure A.206. P_D vs Time

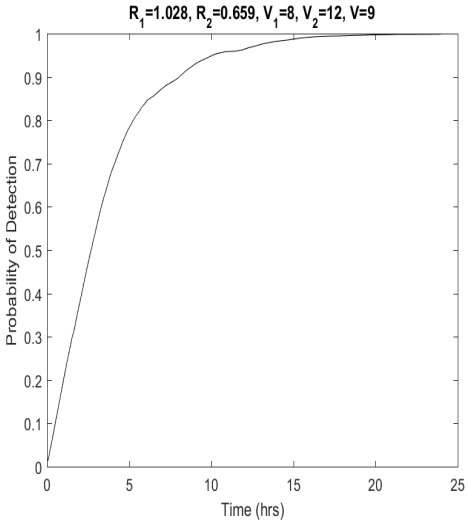


Figure A.207. P_D vs Time

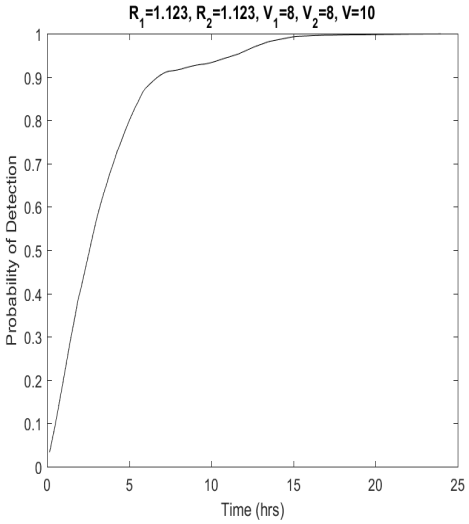


Figure A.208. P_D vs Time

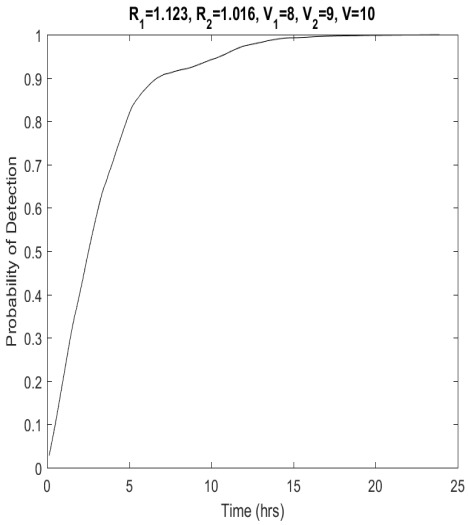


Figure A.209. P_D vs Time

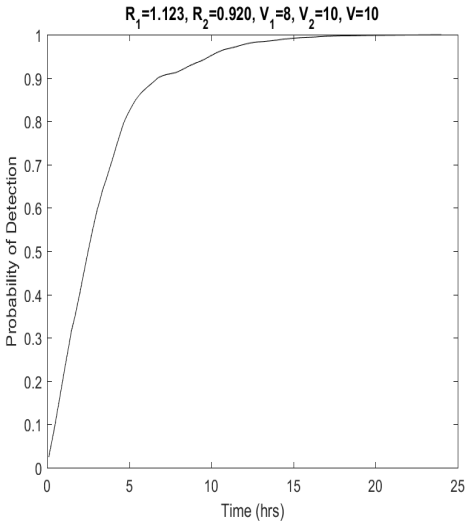


Figure A.210. P_D vs Time

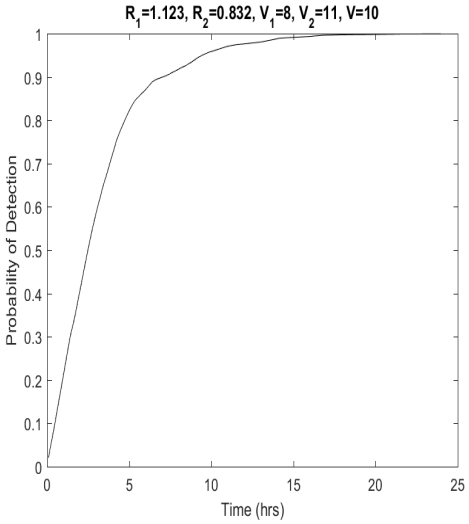


Figure A.211. P_D vs Time

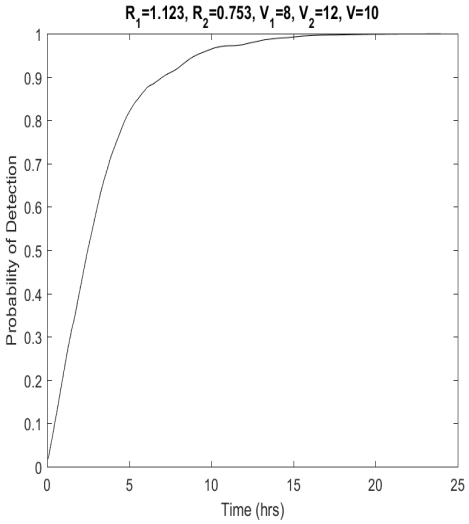


Figure A.212. P_D vs Time

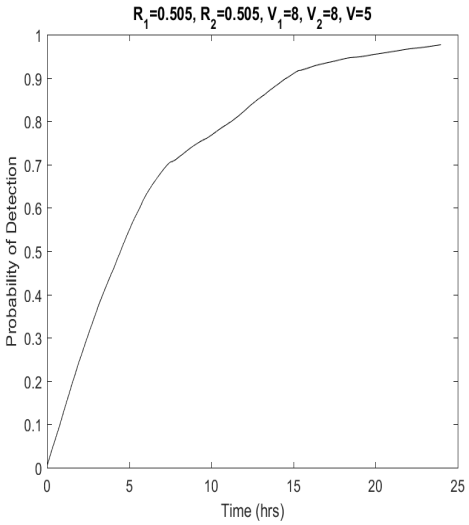


Figure A.213. P_D vs Time

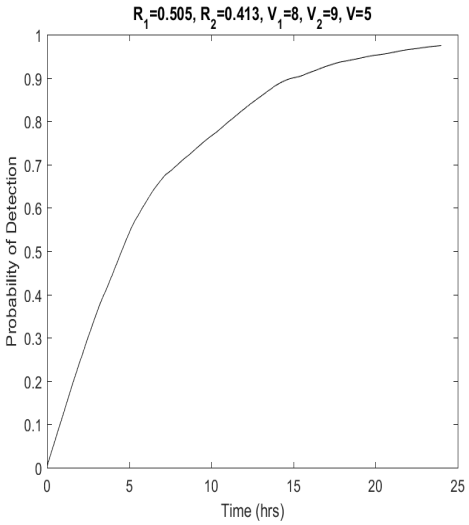


Figure A.214. P_D vs Time

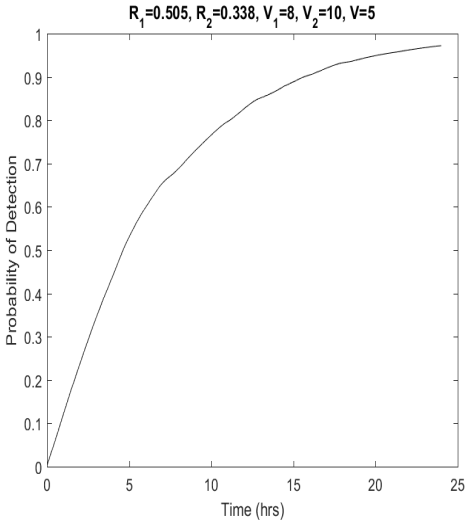


Figure A.215. P_D vs Time

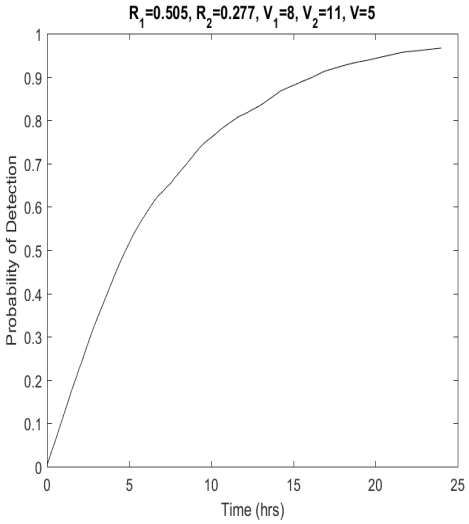


Figure A.216. P_D vs Time

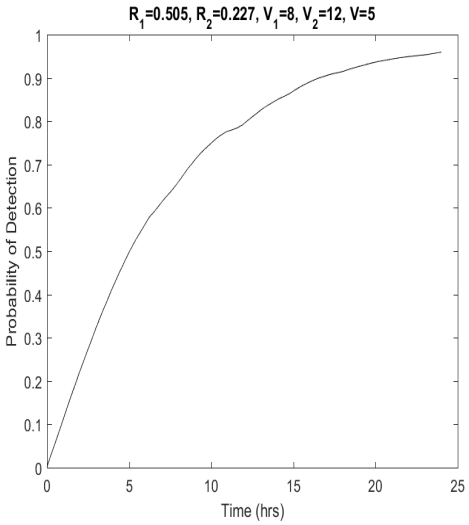


Figure A.217. P_D vs Time

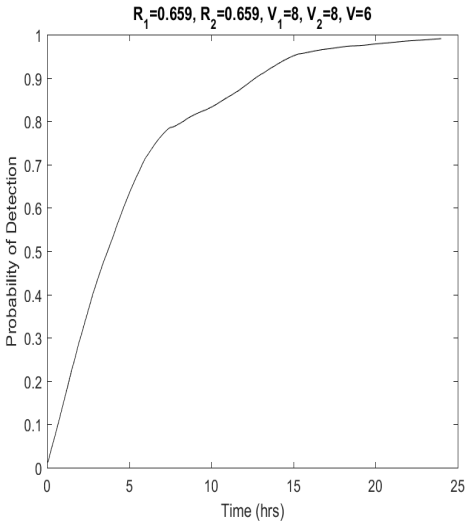


Figure A.218. P_D vs Time

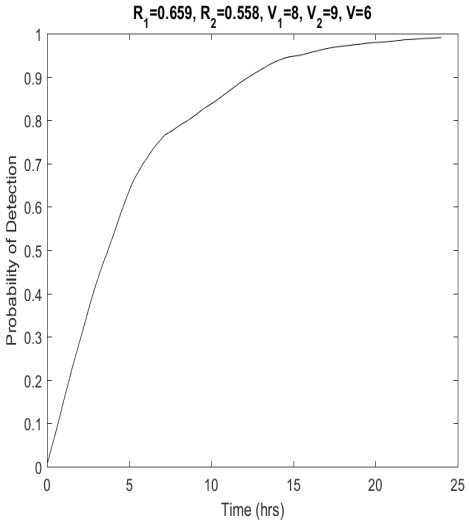


Figure A.219. P_D vs Time

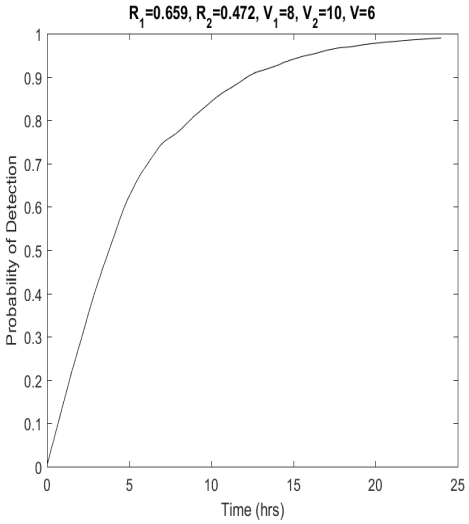


Figure A.220. P_D vs Time

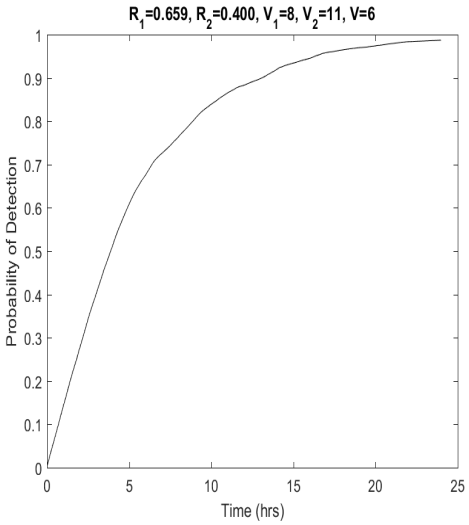


Figure A.221. P_D vs Time

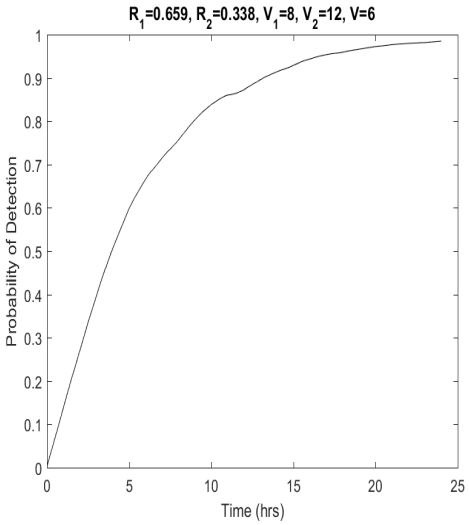


Figure A.222. P_D vs Time

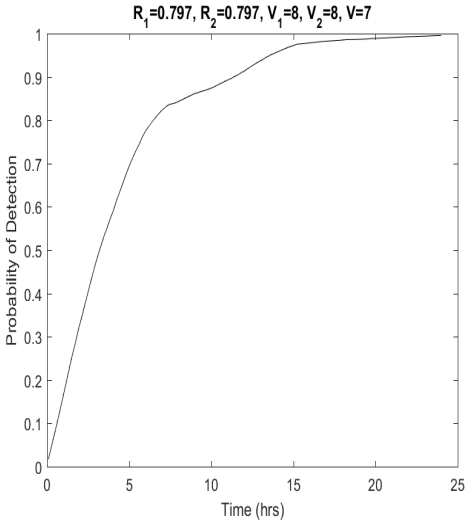


Figure A.223. P_D vs Time

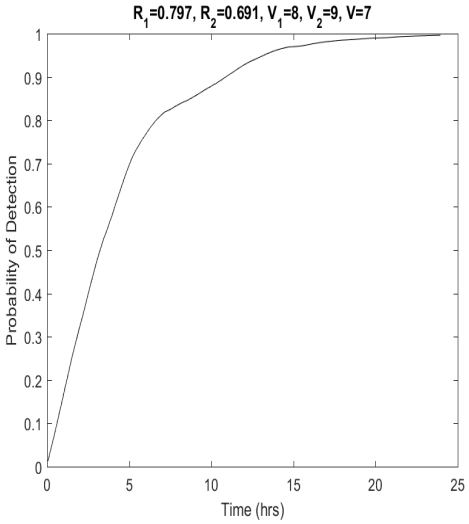


Figure A.224. P_D vs Time

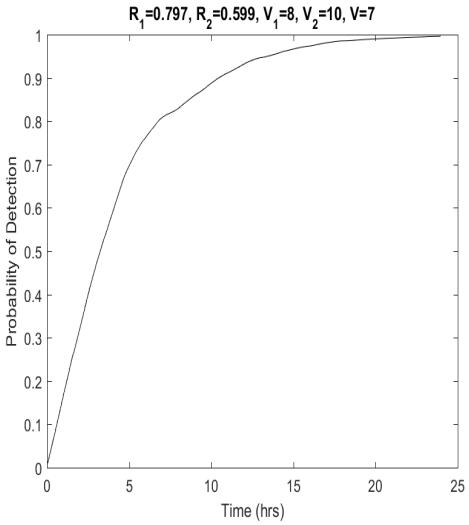


Figure A.225. P_D vs Time

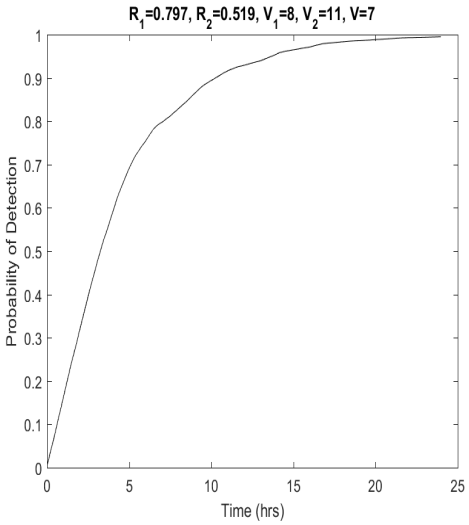


Figure A.226. P_D vs Time

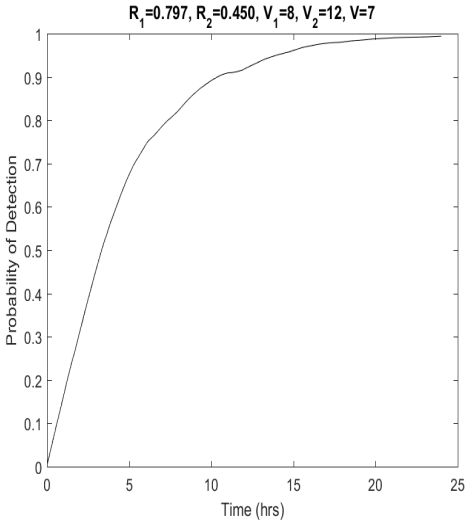


Figure A.227. P_D vs Time

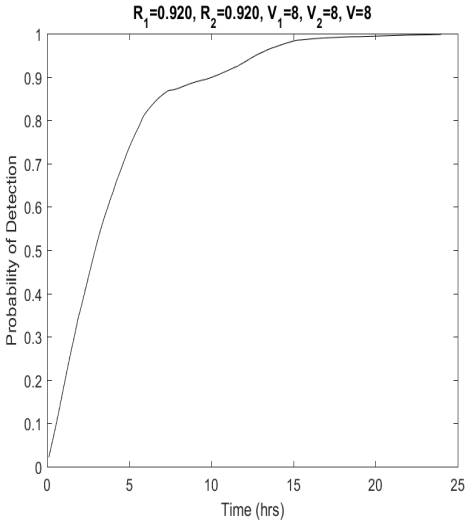


Figure A.228. P_D vs Time

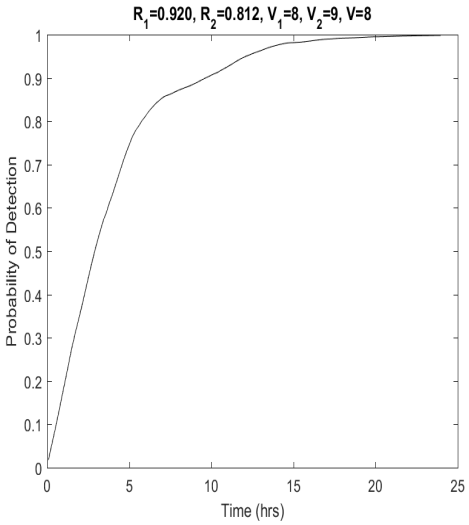


Figure A.229. P_D vs Time

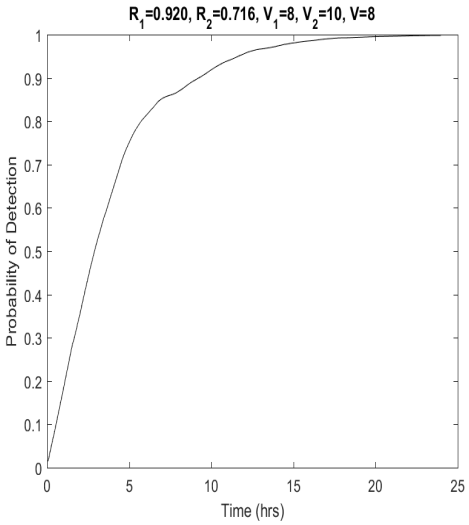


Figure A.230. P_D vs Time

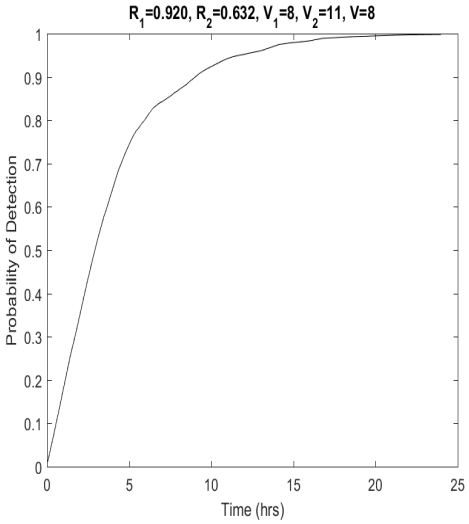


Figure A.231. P_D vs Time

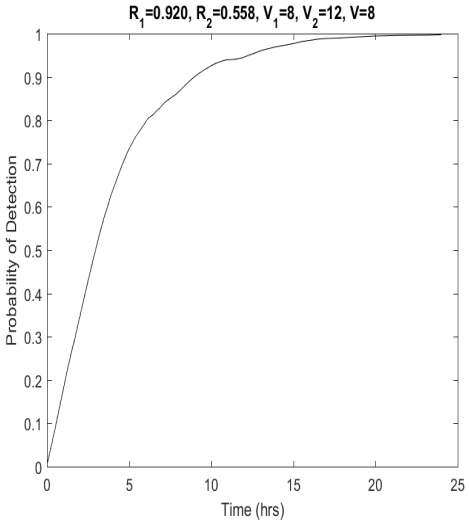


Figure A.232. P_D vs Time

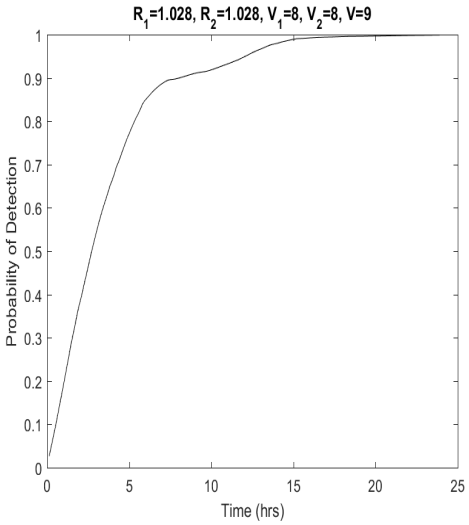


Figure A.233. P_D vs Time

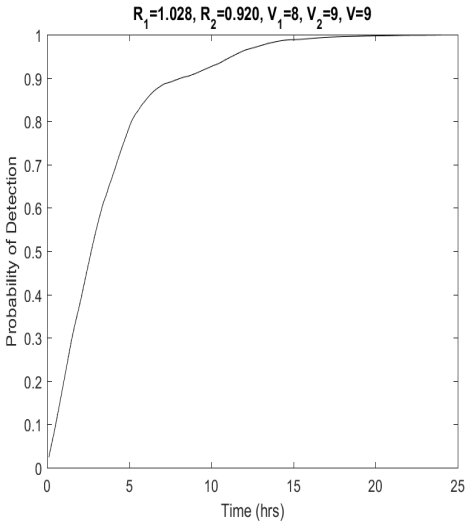


Figure A.234. P_D vs Time

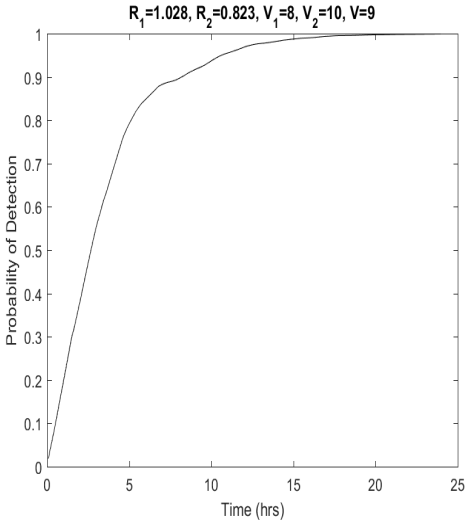


Figure A.235. P_D vs Time

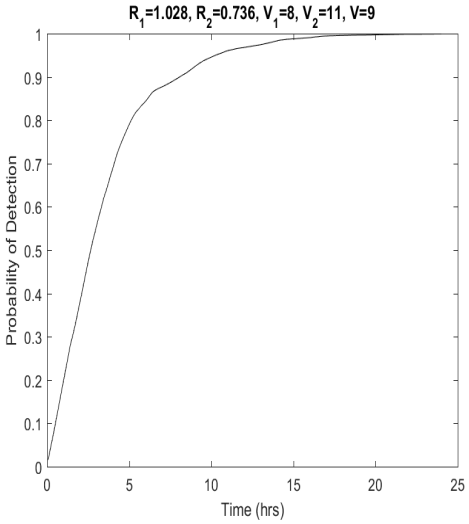


Figure A.236. P_D vs Time

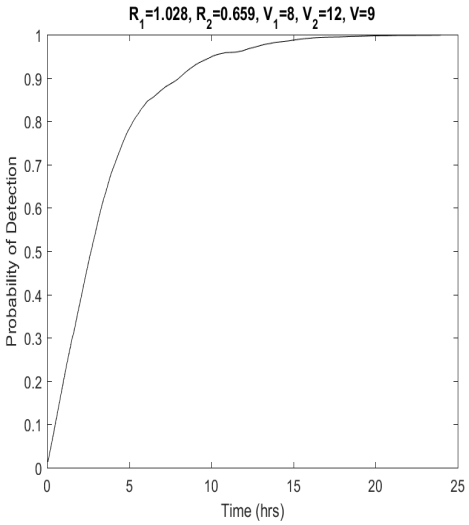


Figure A.237. P_D vs Time

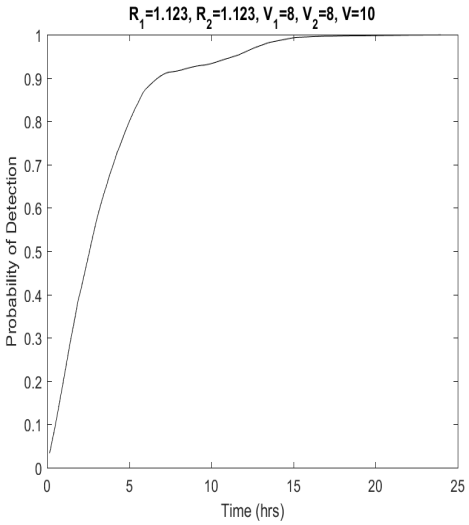


Figure A.238. P_D vs Time

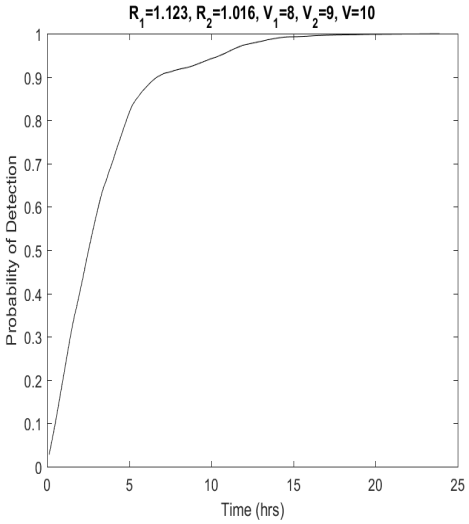


Figure A.239. P_D vs Time

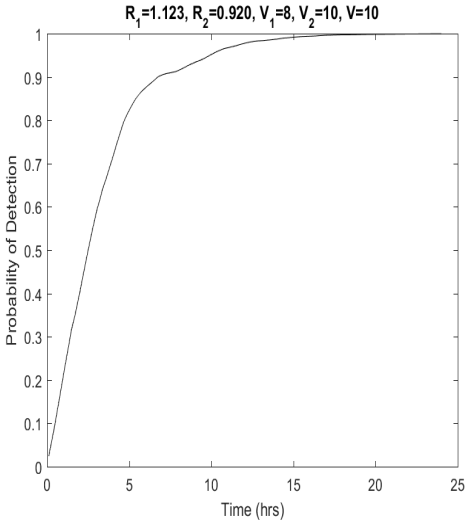


Figure A.240. P_D vs Time

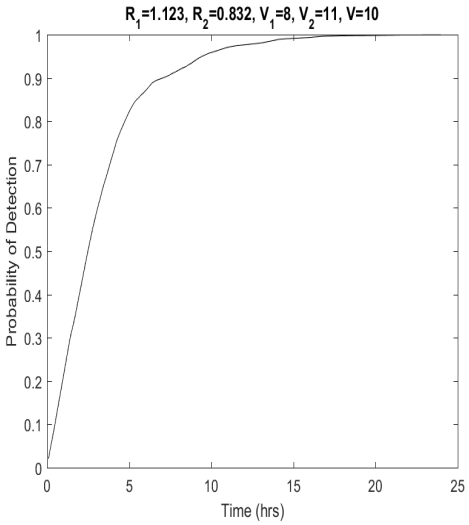


Figure A.241. P_D vs Time

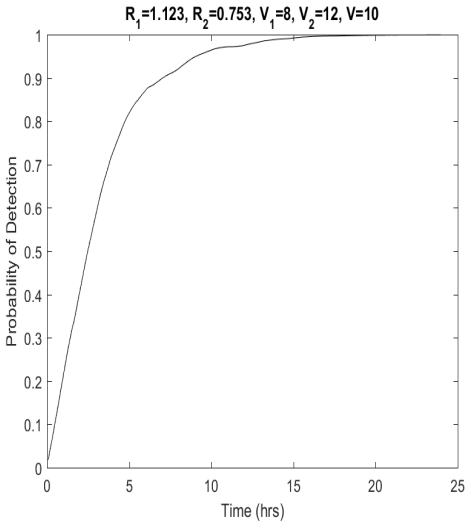


Figure A.242. P_D vs Time

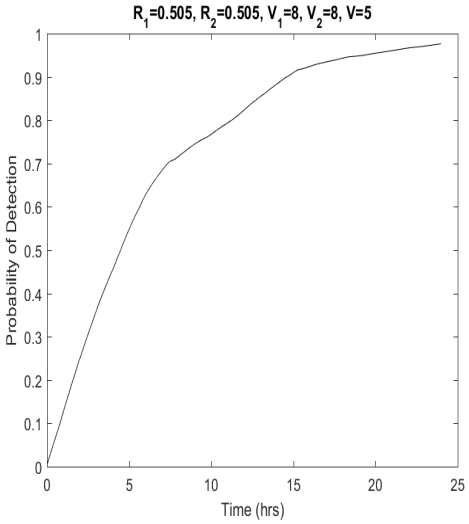


Figure A.243. P_D vs Time

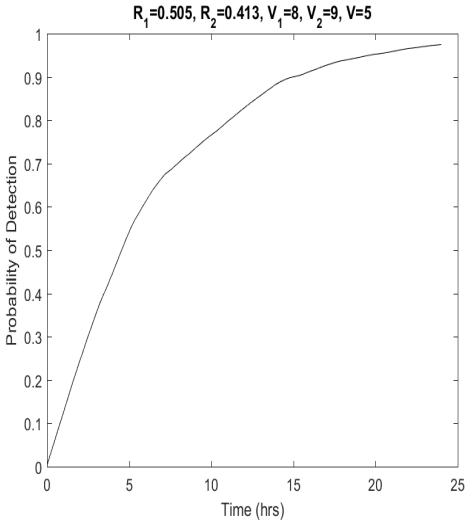


Figure A.244. P_D vs Time

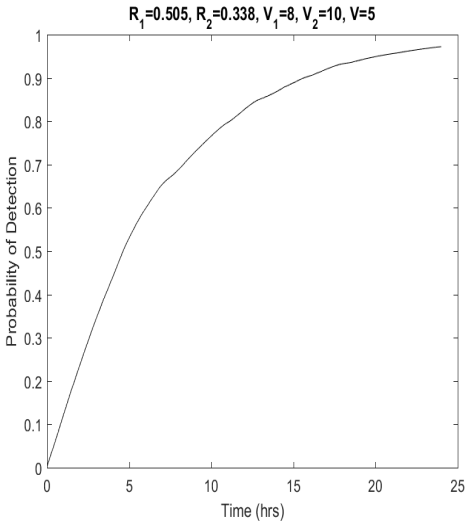


Figure A.245. P_D vs Time

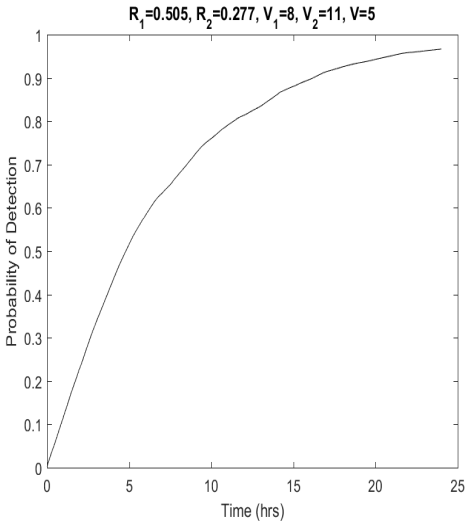


Figure A.246. P_D vs Time

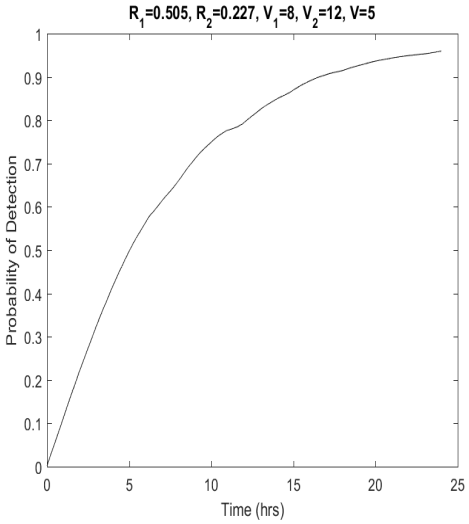


Figure A.247. P_D vs Time

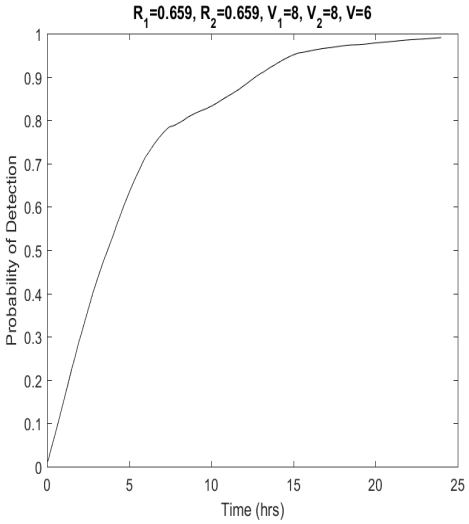


Figure A.248. P_D vs Time

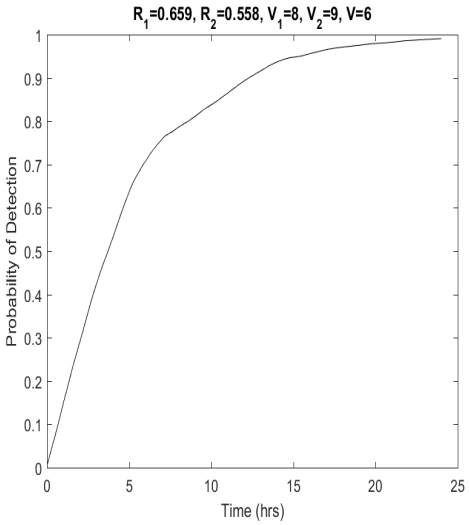


Figure A.249. P_D vs Time

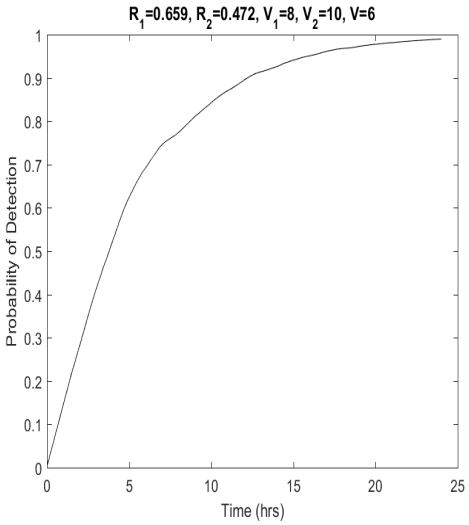


Figure A.250. P_D vs Time

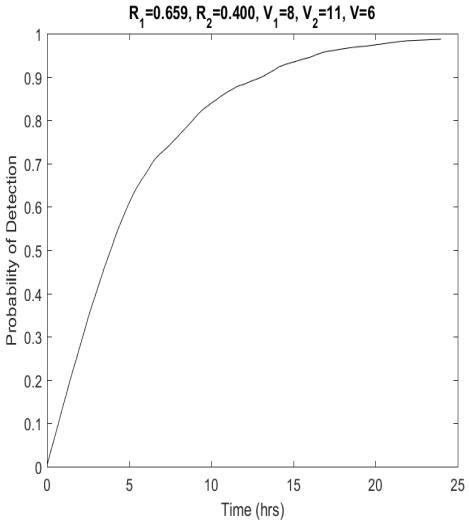


Figure A.251. P_D vs Time

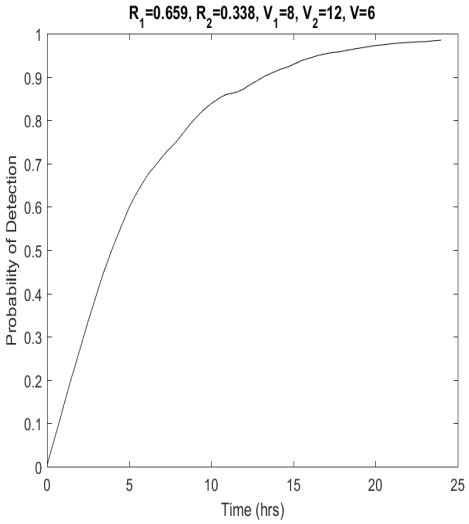


Figure A.252. P_D vs Time

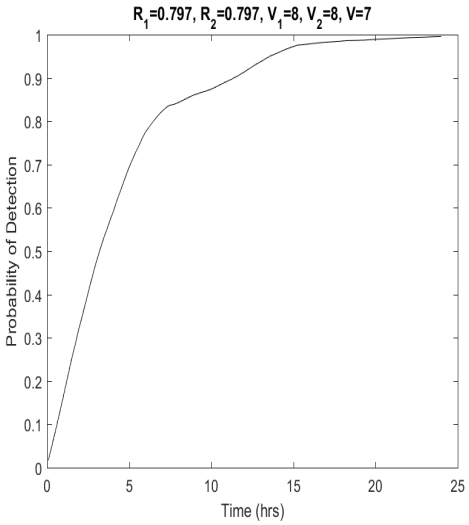


Figure A.253. P_D vs Time

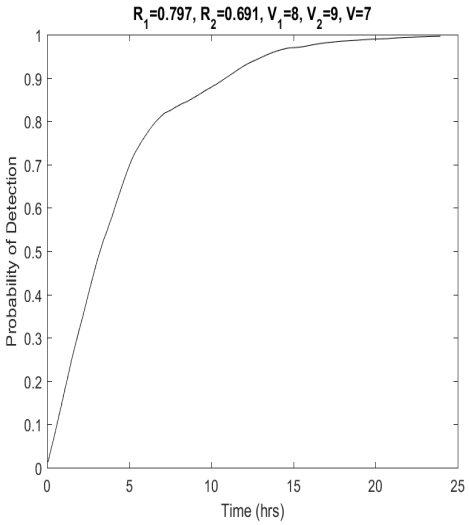


Figure A.254. P_D vs Time

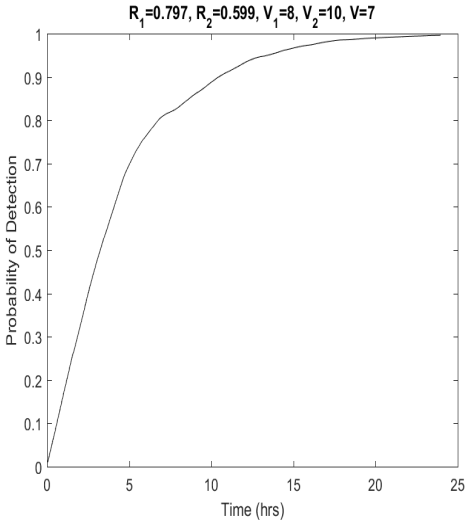


Figure A.255. P_D vs Time

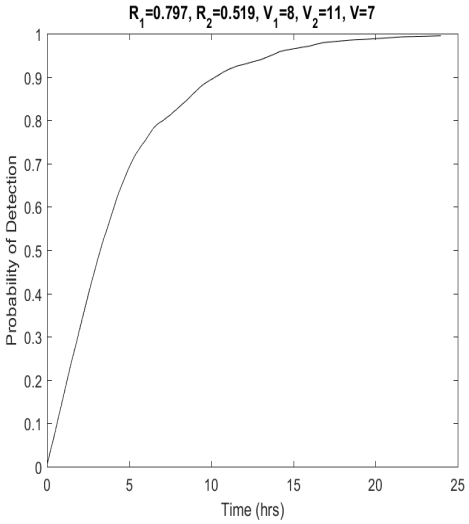


Figure A.256. P_D vs Time

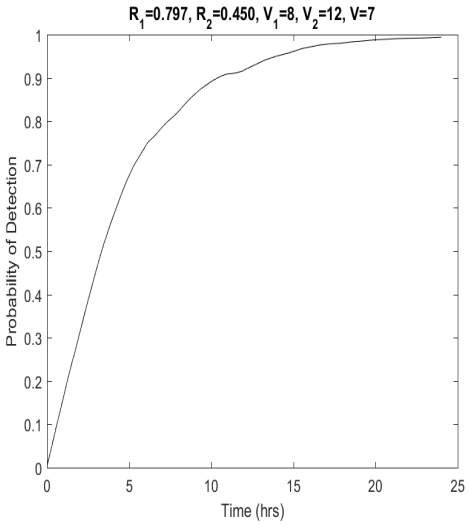


Figure A.257. P_D vs Time

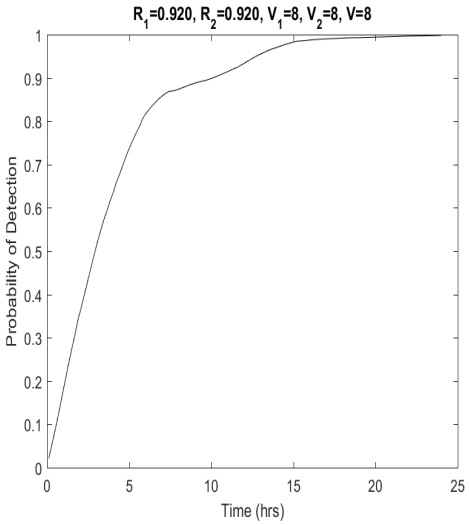


Figure A.258. P_D vs Time

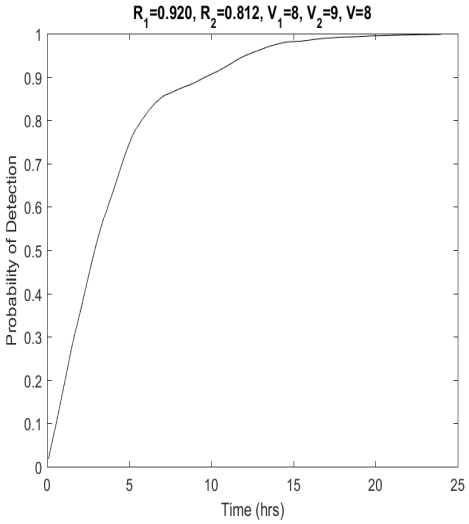


Figure A.259. P_D vs Time

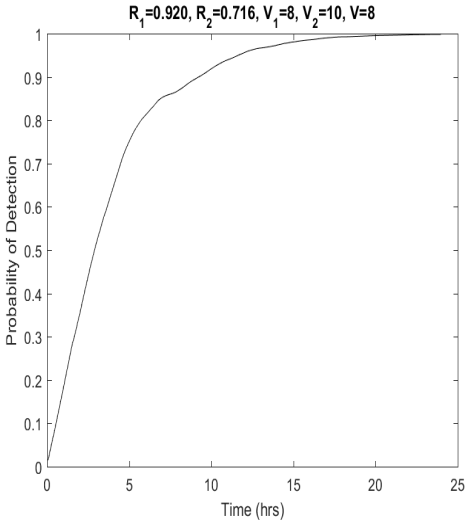


Figure A.260. P_D vs Time

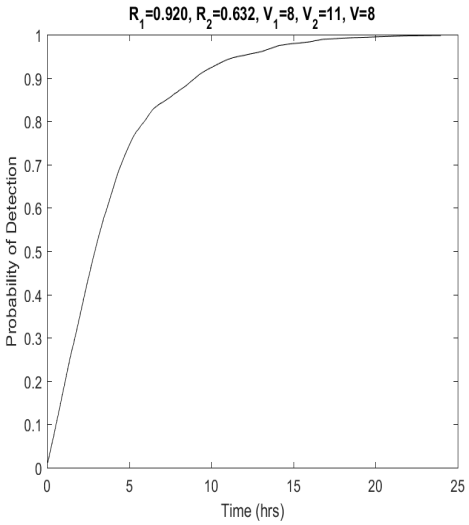


Figure A.261. P_D vs Time

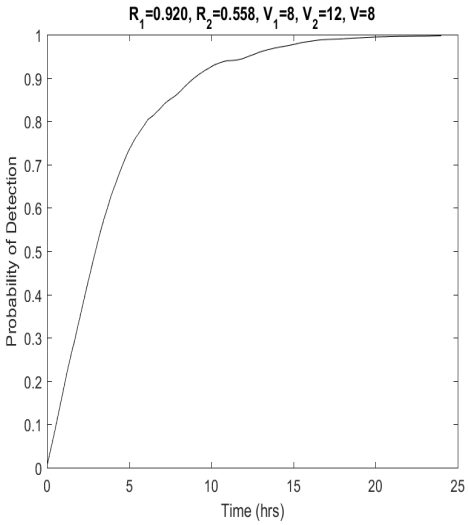


Figure A.262. P_D vs Time

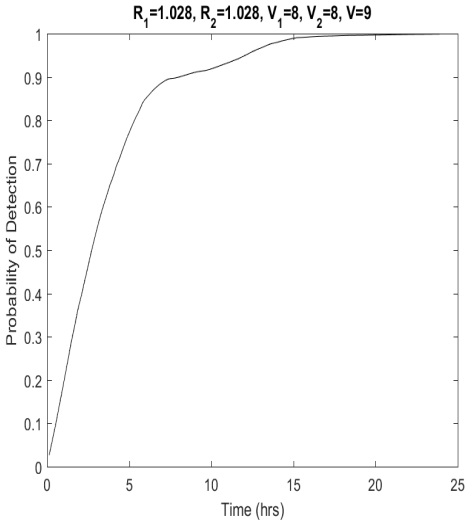


Figure A.263. P_D vs Time

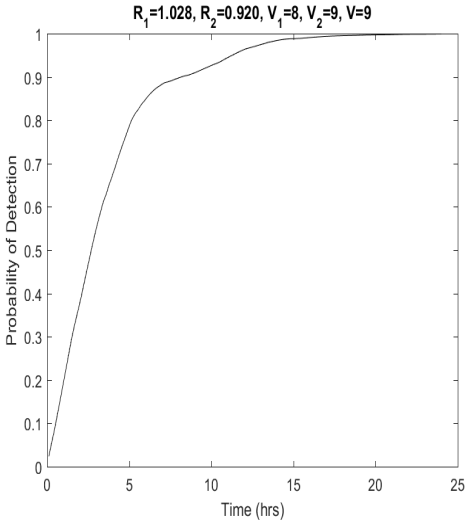


Figure A.264. P_D vs Time

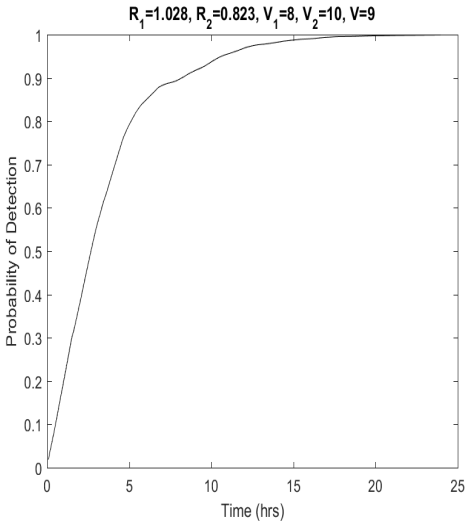


Figure A.265. P_D vs Time

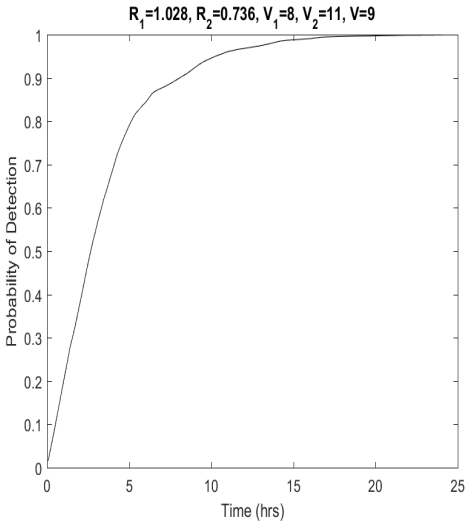


Figure A.266. P_D vs Time

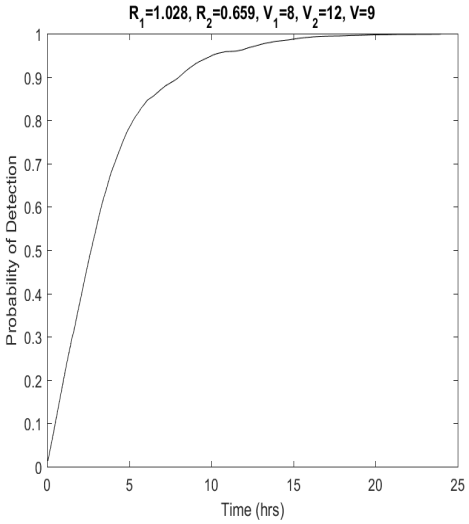


Figure A.267. P_D vs Time

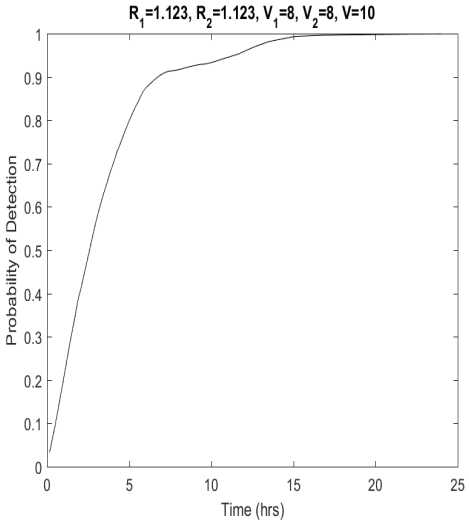


Figure A.268. P_D vs Time

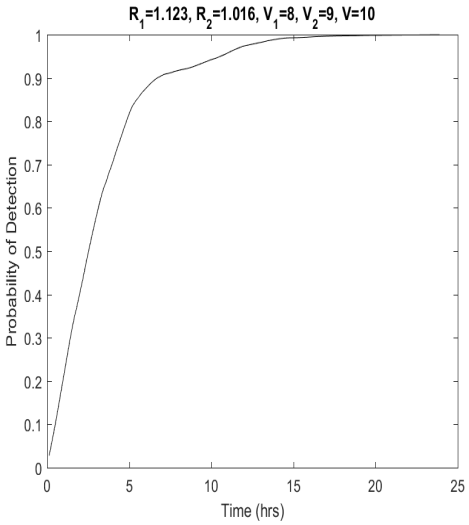


Figure A.269. P_D vs Time

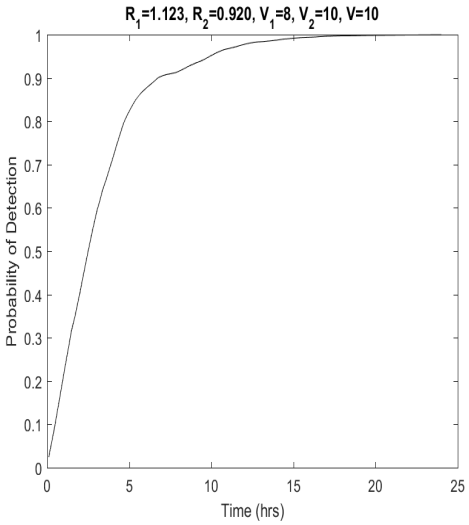


Figure A.270. P_D vs Time

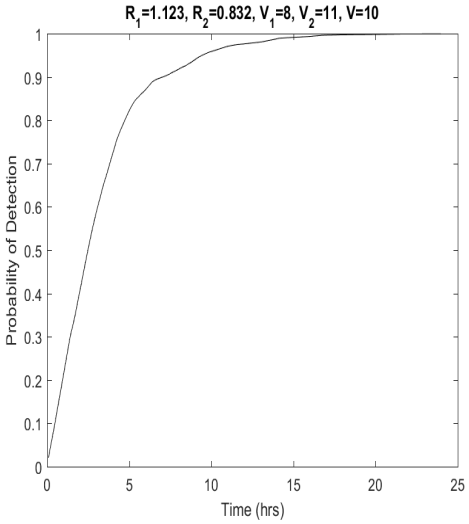


Figure A.271. P_D vs Time

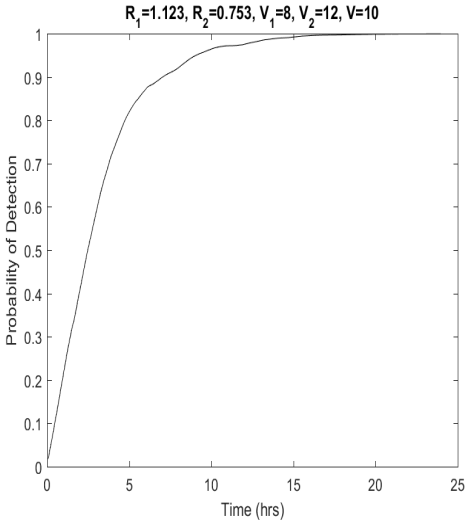


Figure A.272. P_D vs Time

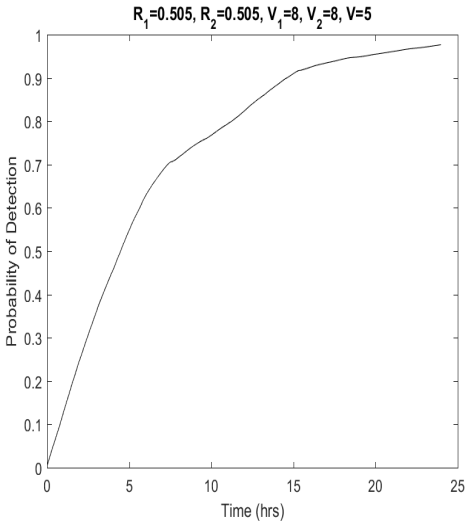


Figure A.273. P_D vs Time

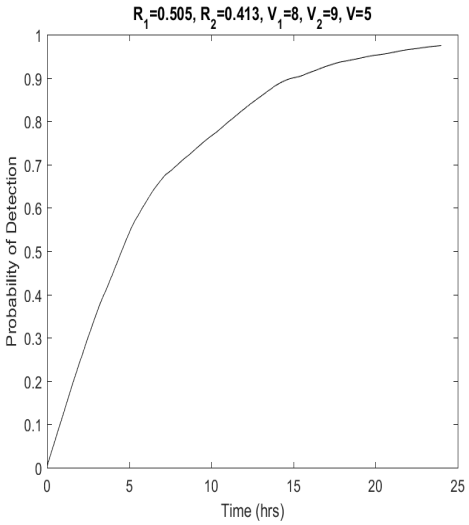


Figure A.274. P_D vs Time

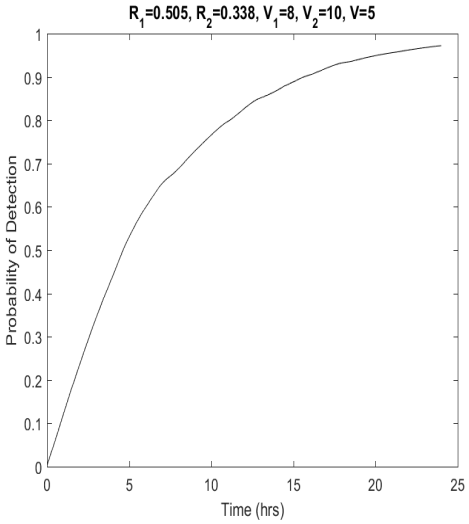


Figure A.275. P_D vs Time

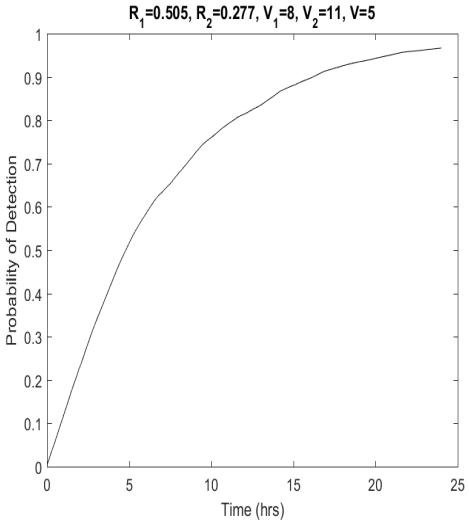


Figure A.276. P_D vs Time

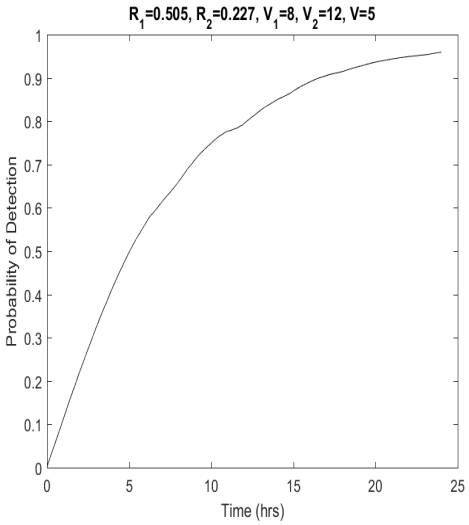


Figure A.277. P_D vs Time

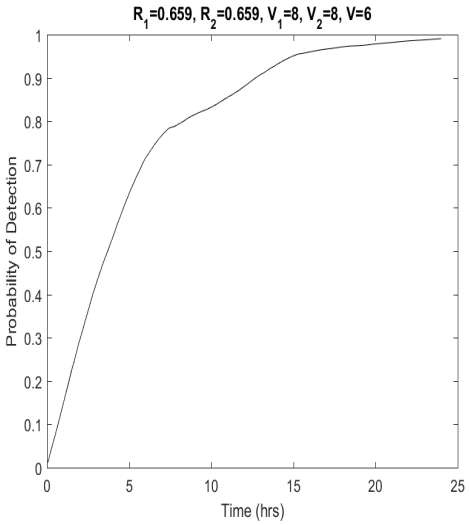


Figure A.278. P_D vs Time

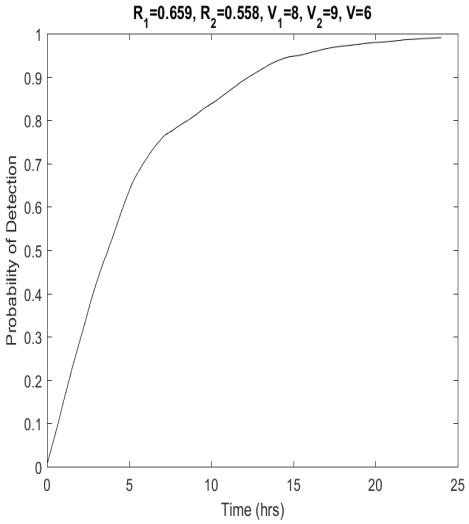


Figure A.279. P_D vs Time

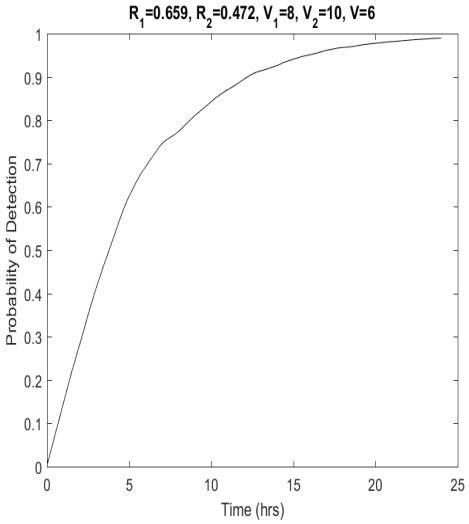


Figure A.280. P_D vs Time

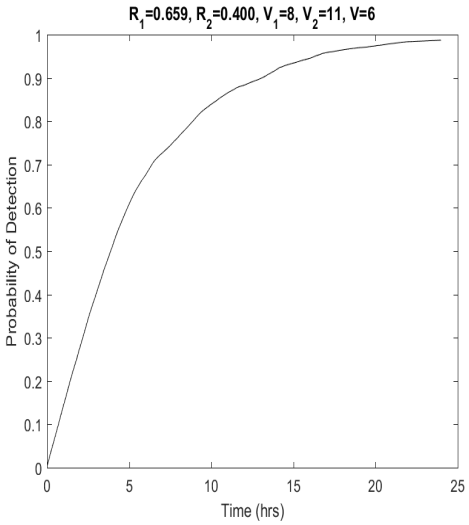


Figure A.281. P_D vs Time

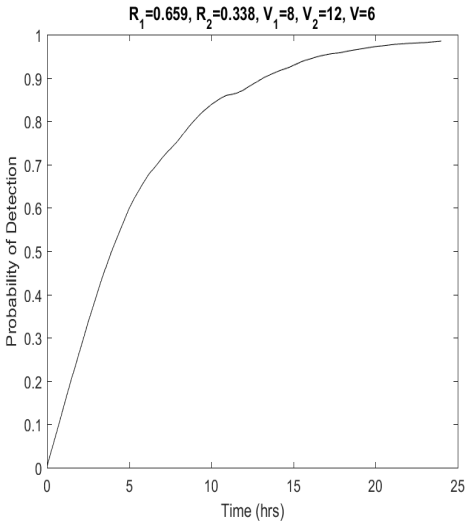


Figure A.282. P_D vs Time

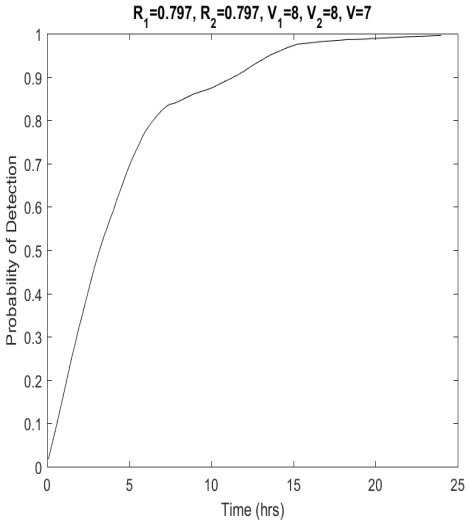


Figure A.283. P_D vs Time

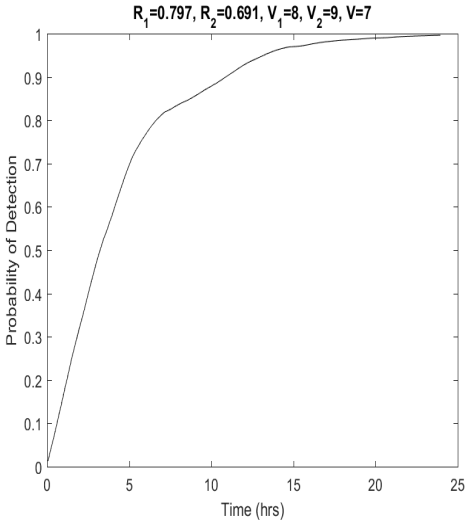


Figure A.284. P_D vs Time

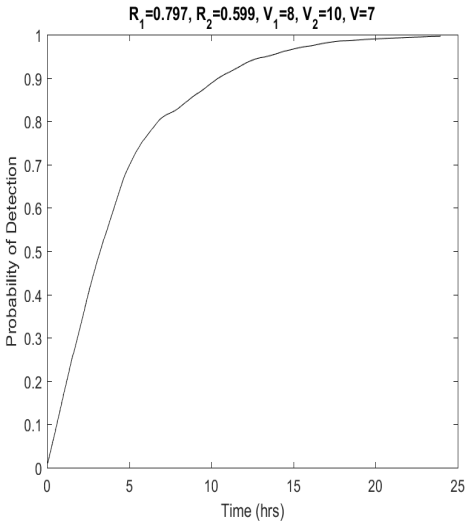


Figure A.285. P_D vs Time

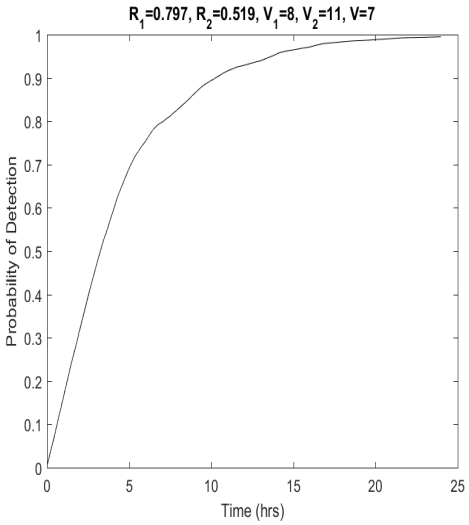


Figure A.286. P_D vs Time

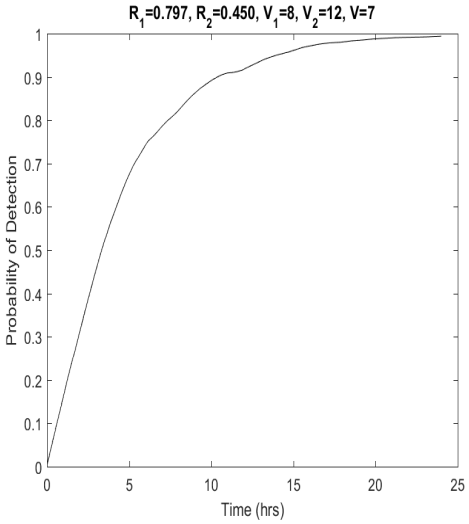


Figure A.287. P_D vs Time

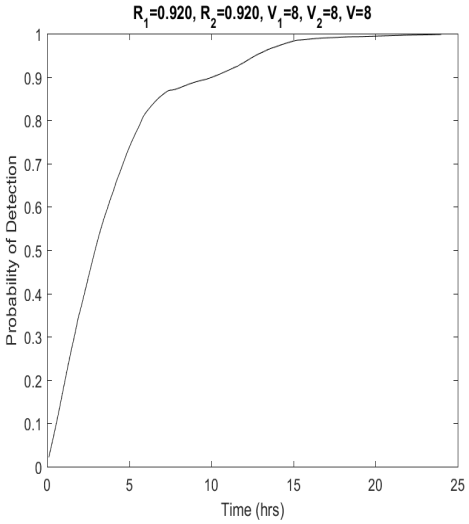


Figure A.288. P_D vs Time

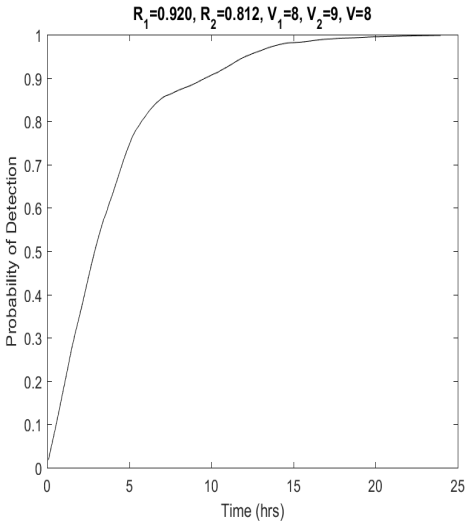


Figure A.289. P_D vs Time

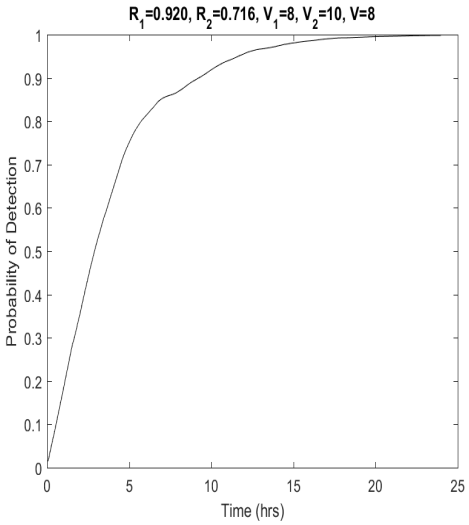


Figure A.290. P_D vs Time

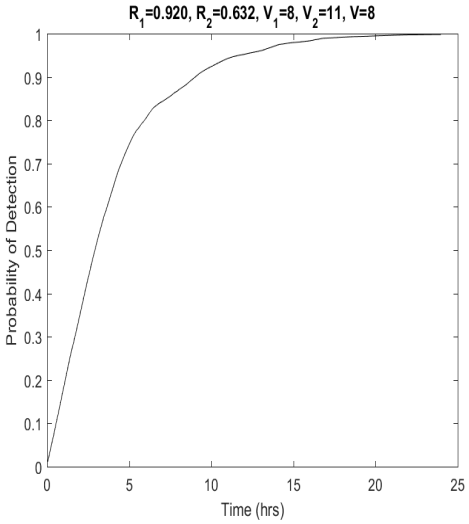


Figure A.291. P_D vs Time

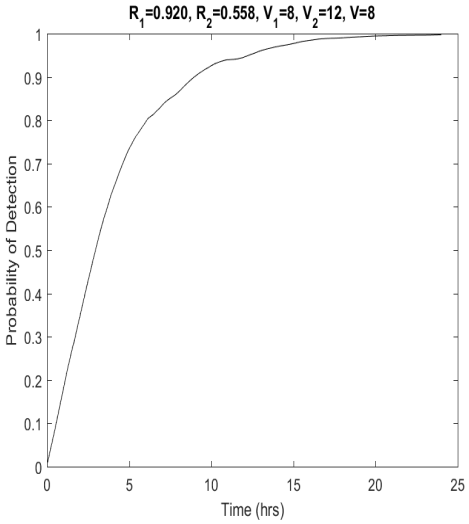


Figure A.292. P_D vs Time

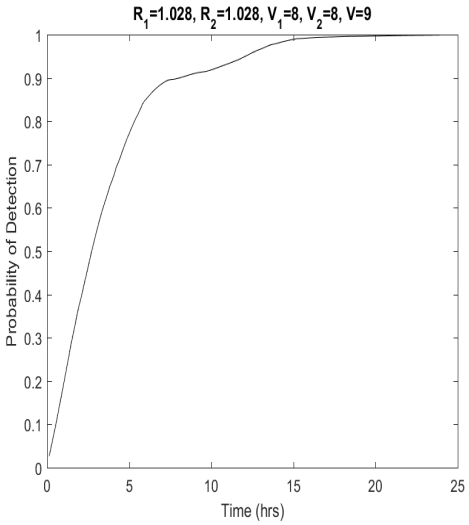


Figure A.293. P_D vs Time

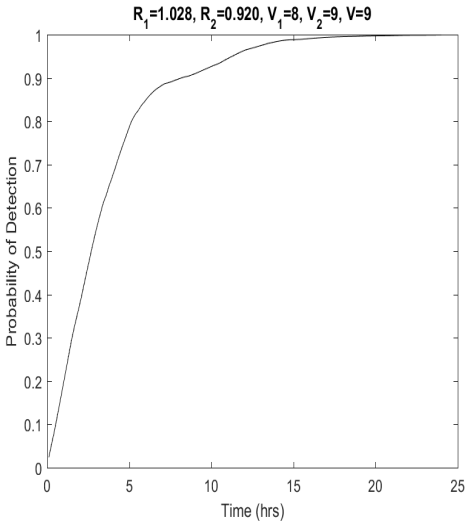


Figure A.294. P_D vs Time

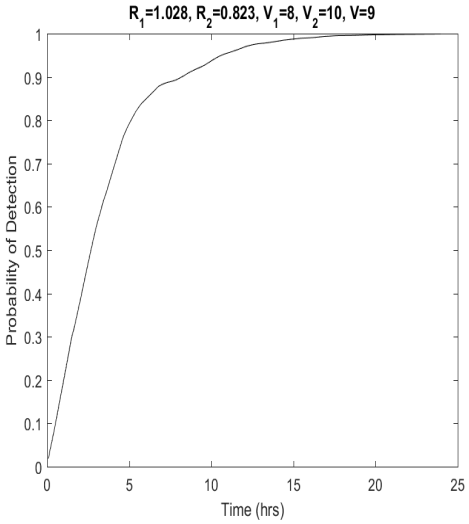


Figure A.295. P_D vs Time

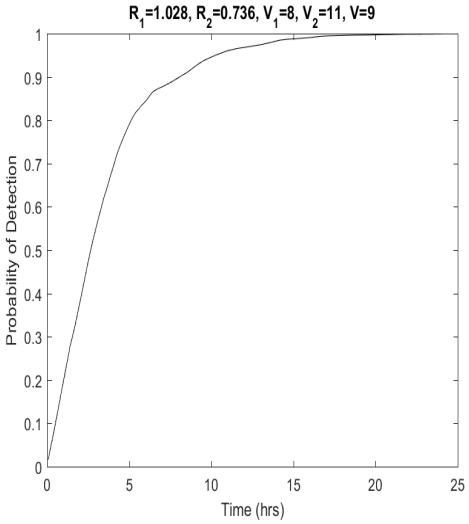


Figure A.296. P_D vs Time

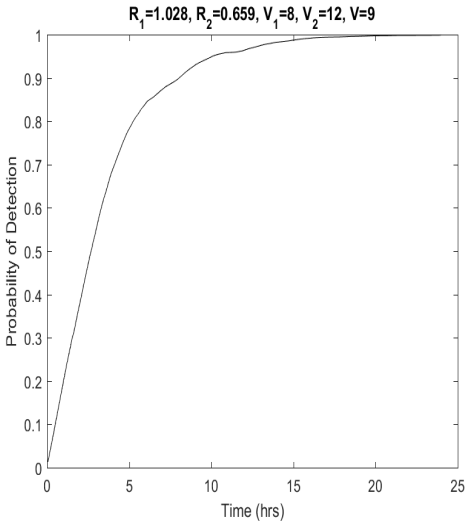


Figure A.297. P_D vs Time

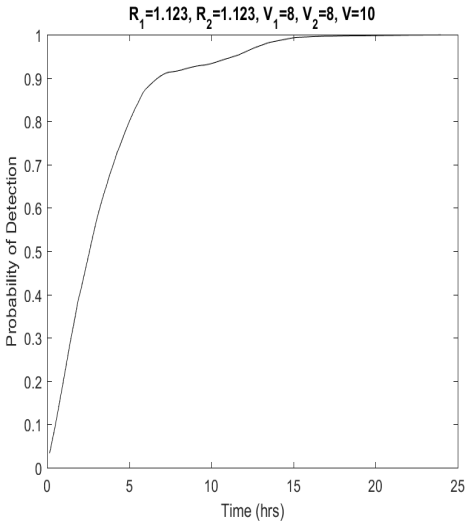


Figure A.298. P_D vs Time

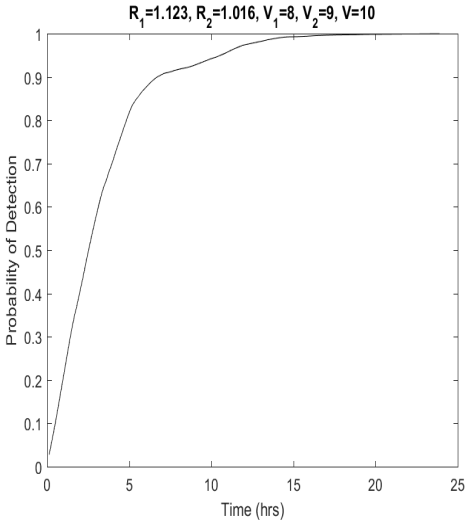


Figure A.299. P_D vs Time

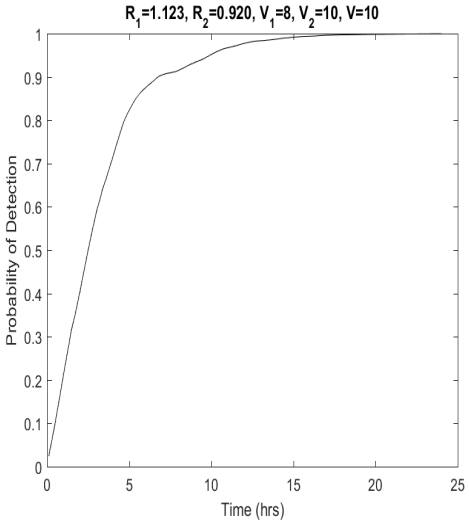


Figure A.300. P_D vs Time

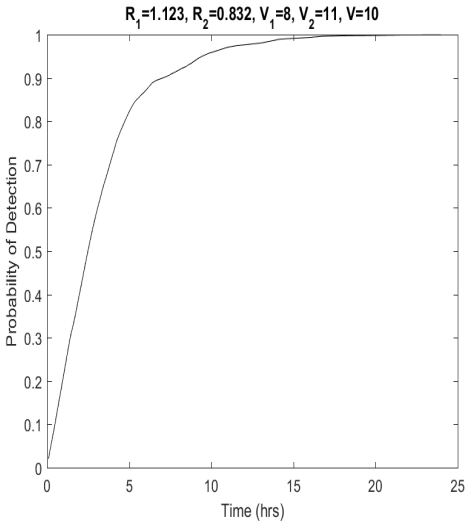


Figure A.301. P_D vs Time

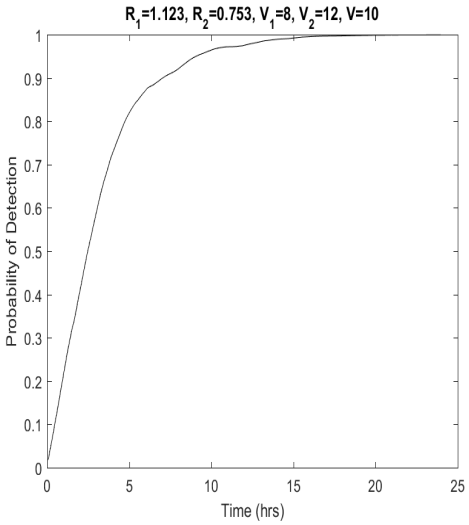


Figure A.302. P_D vs Time

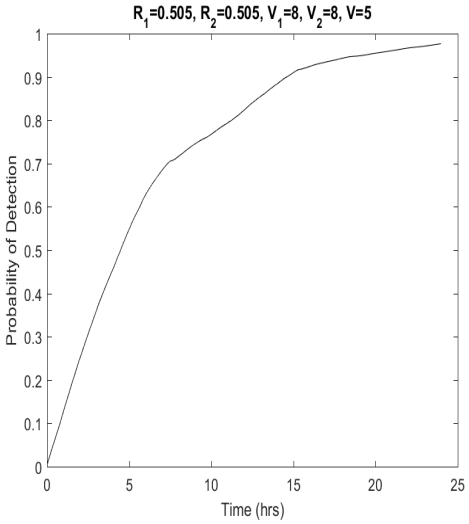


Figure A.303. P_D vs Time

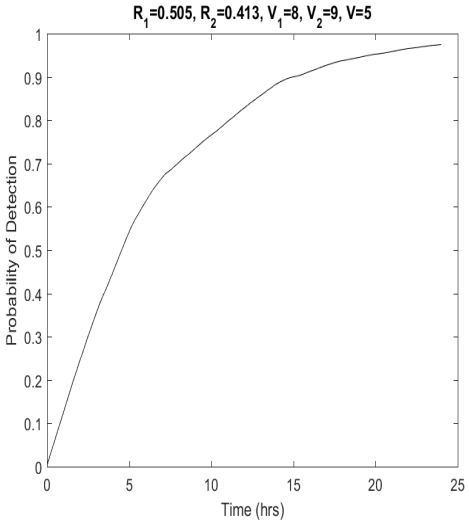


Figure A.304. P_D vs Time

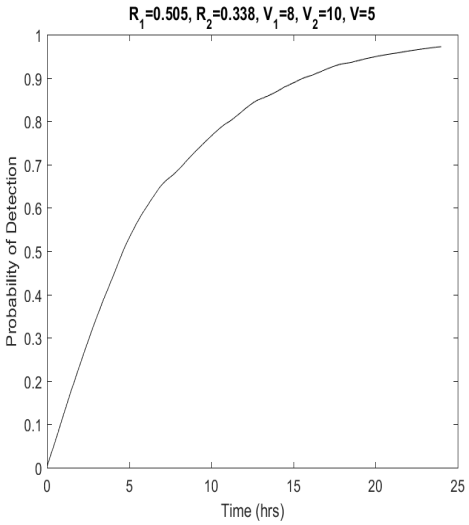


Figure A.305. P_D vs Time

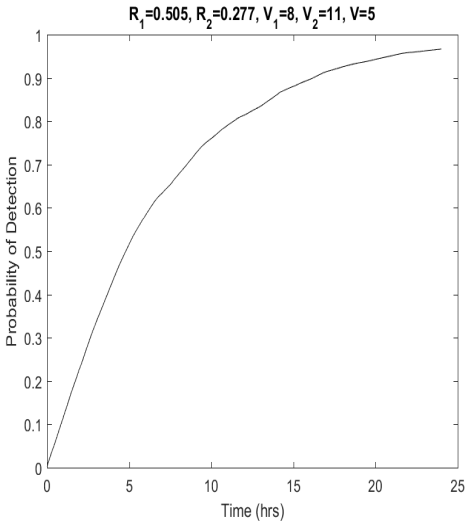


Figure A.306. P_D vs Time

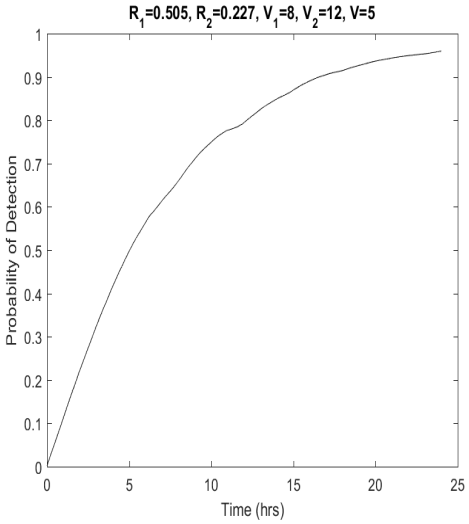


Figure A.307. P_D vs Time

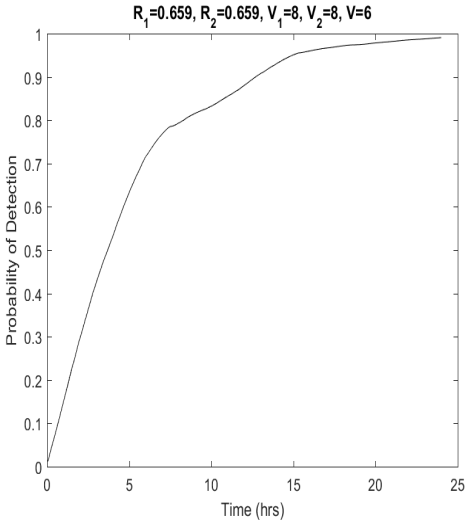


Figure A.308. P_D vs Time

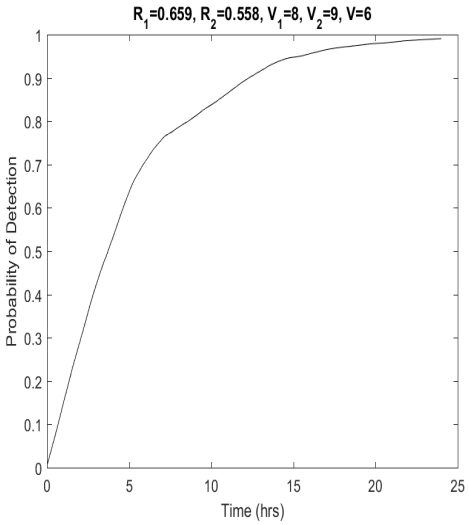


Figure A.309. P_D vs Time

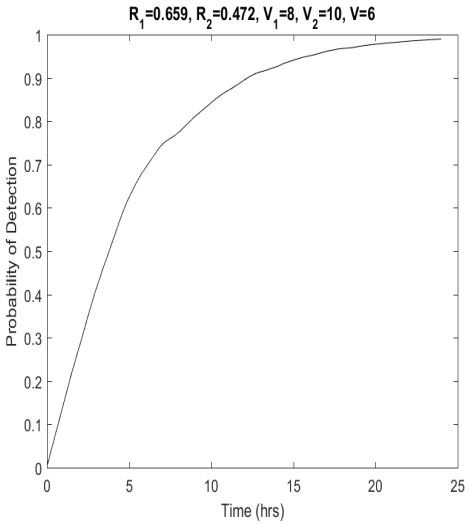


Figure A.310. P_D vs Time

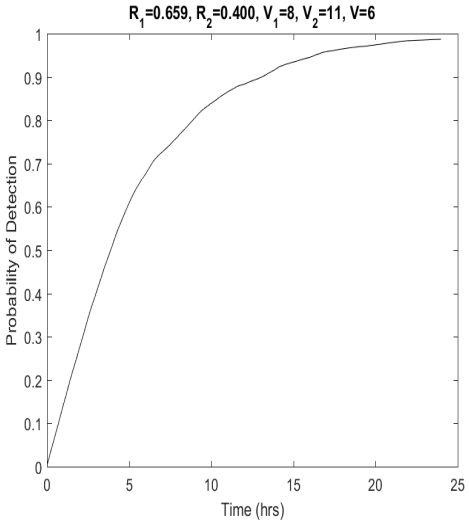


Figure A.311. P_D vs Time

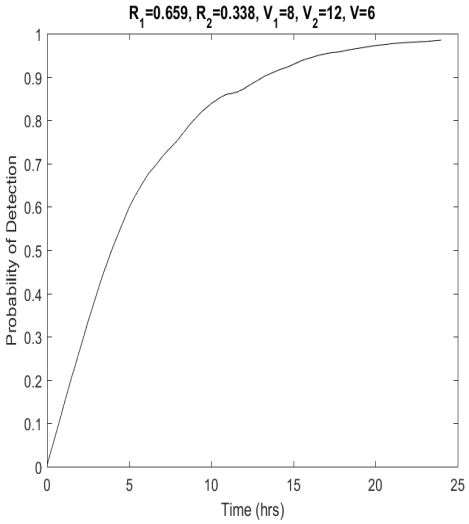


Figure A.312. P_D vs Time

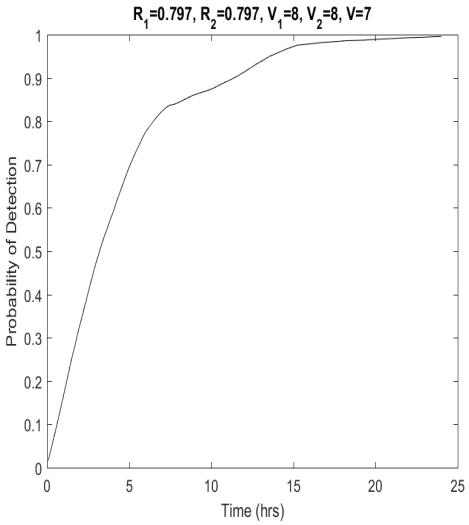


Figure A.313. P_D vs Time

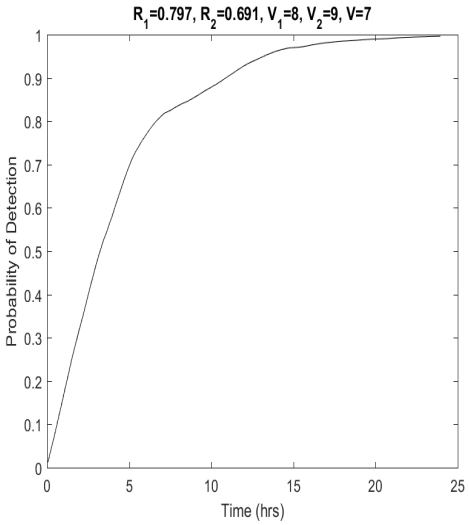


Figure A.314. P_D vs Time

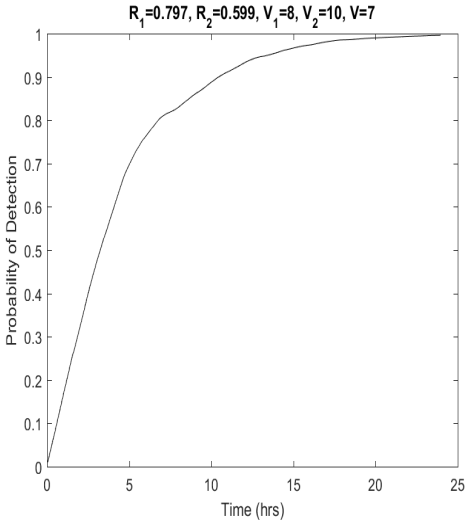


Figure A.315. P_D vs Time

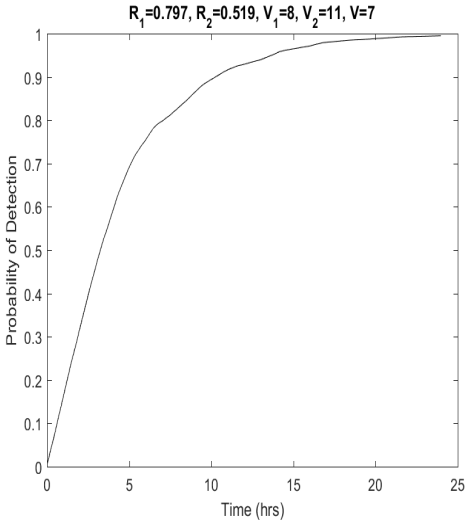


Figure A.316. P_D vs Time

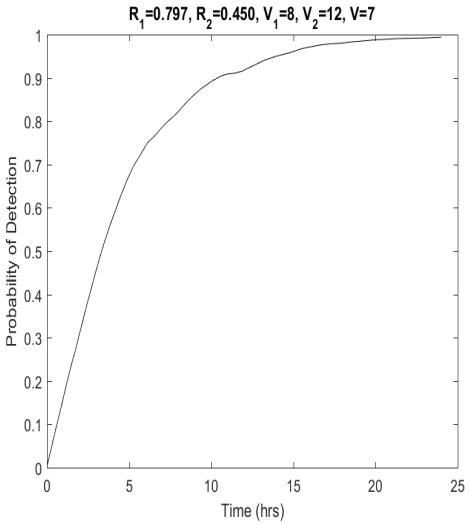


Figure A.317. P_D vs Time

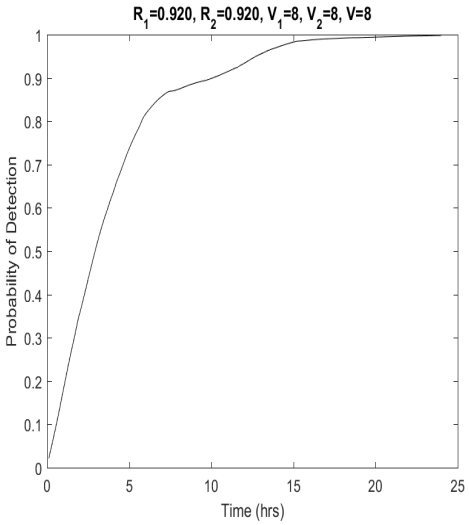


Figure A.318. P_D vs Time

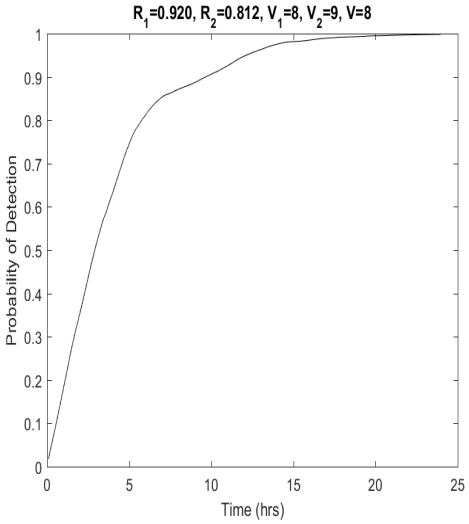


Figure A.319. P_D vs Time

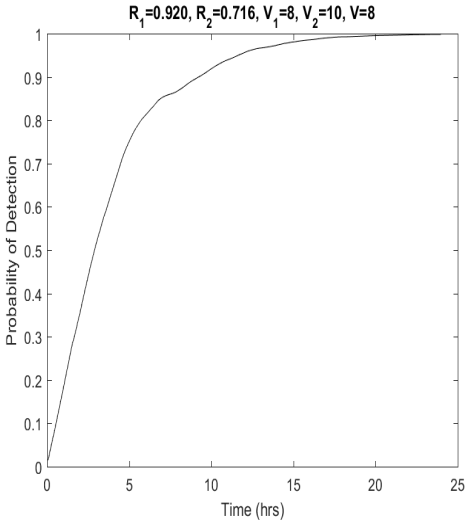


Figure A.320. P_D vs Time

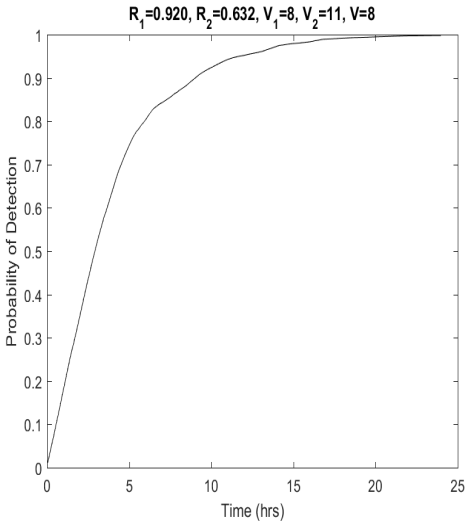


Figure A.321. P_D vs Time

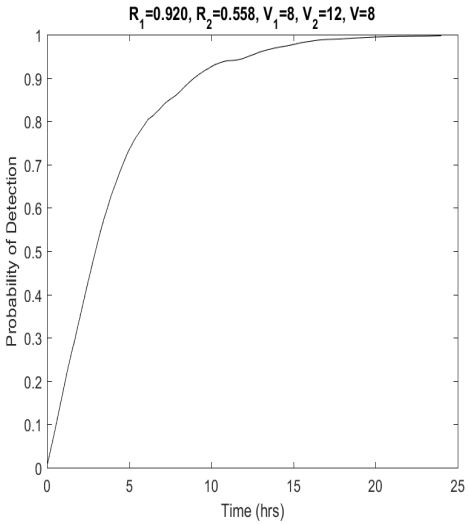


Figure A.322. P_D vs Time

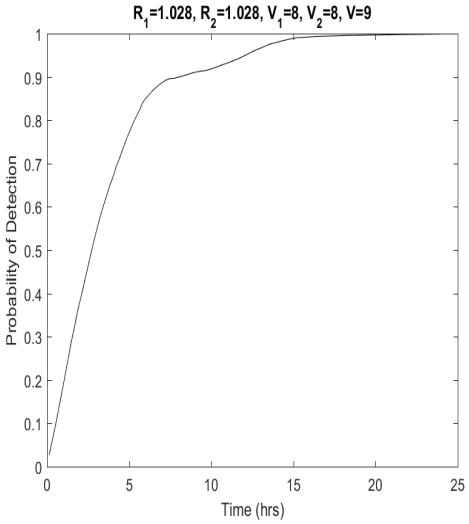


Figure A.323. P_D vs Time

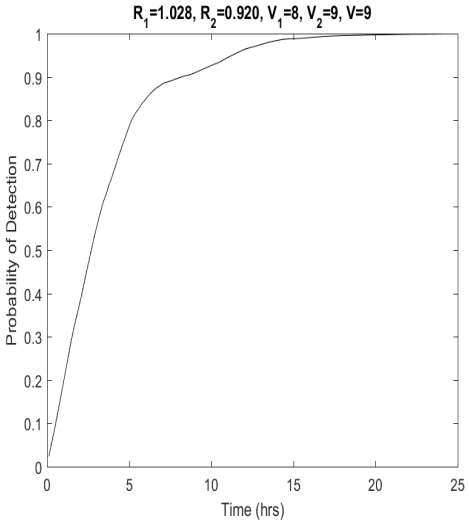


Figure A.324. P_D vs Time

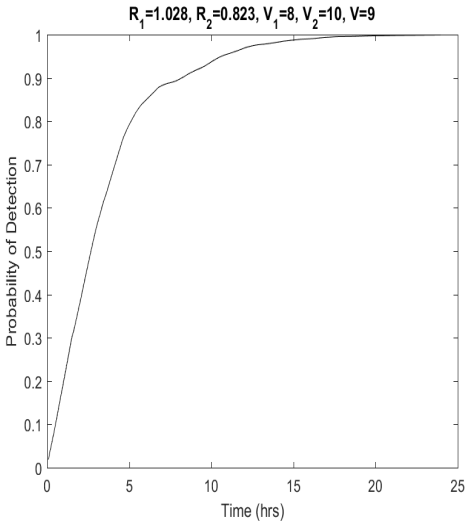


Figure A.325. P_D vs Time

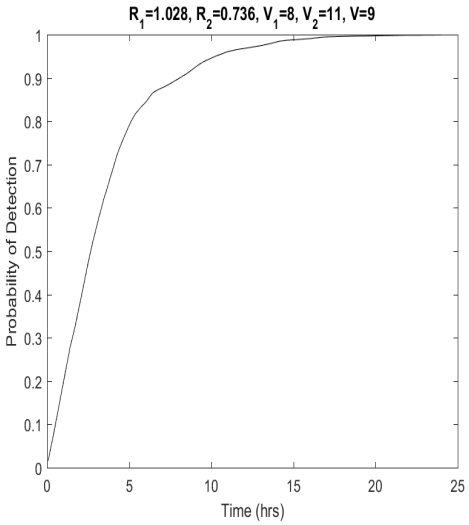


Figure A.326. P_D vs Time

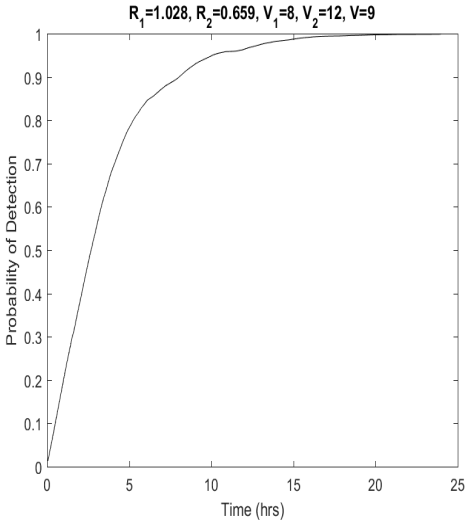


Figure A.327. P_D vs Time

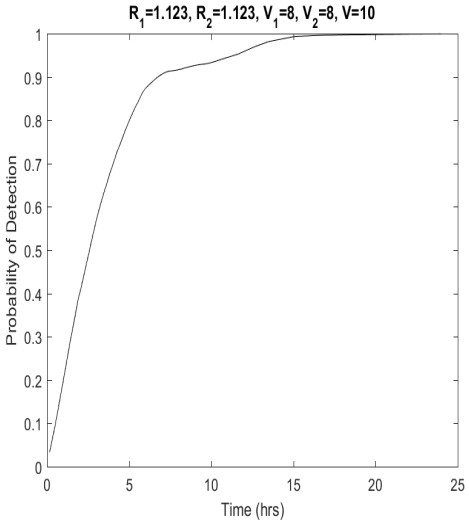


Figure A.328. P_D vs Time

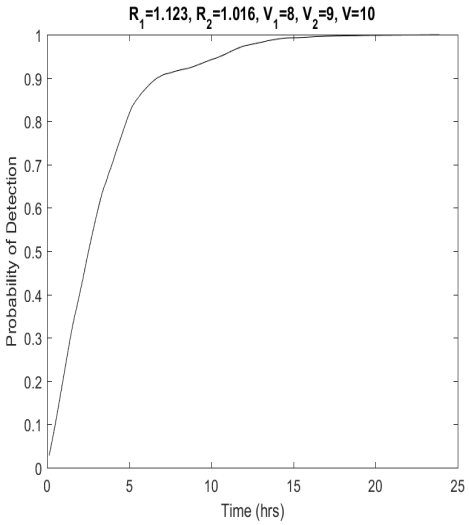


Figure A.329. P_D vs Time

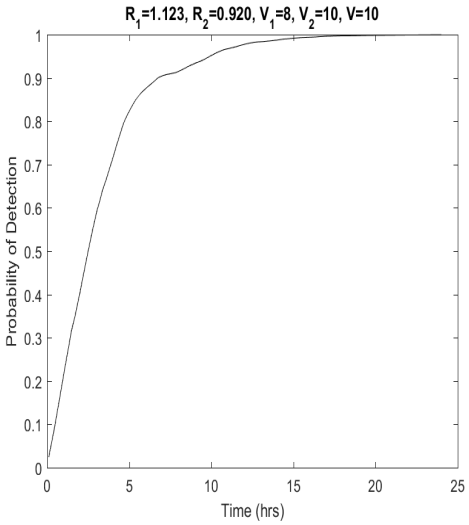


Figure A.330. P_D vs Time

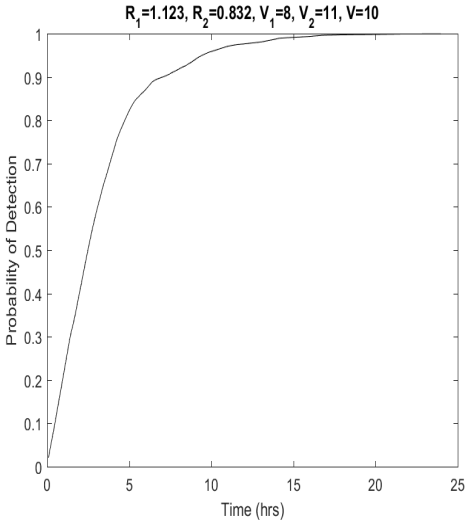


Figure A.331. P_D vs Time

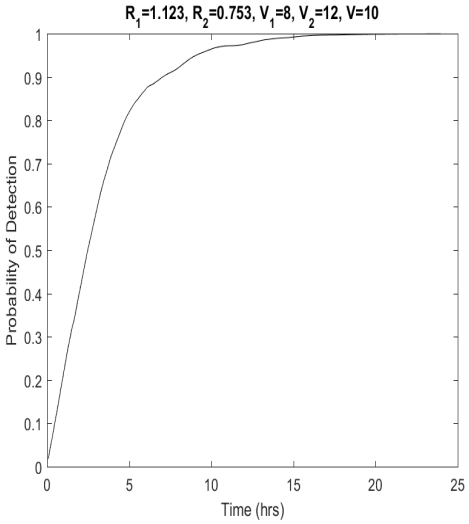


Figure A.332. P_D vs Time

Spiral

A.3.3 Spiral Search

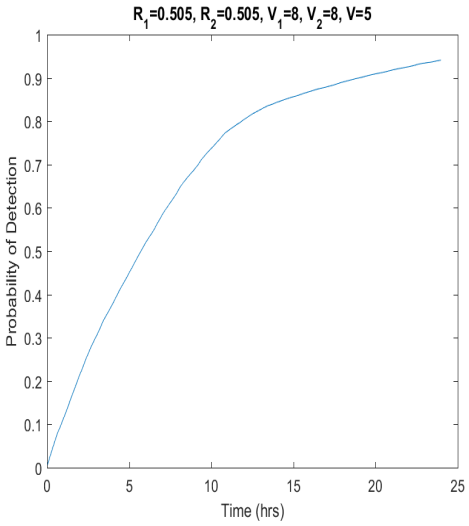


Figure A.333. P_D vs Time

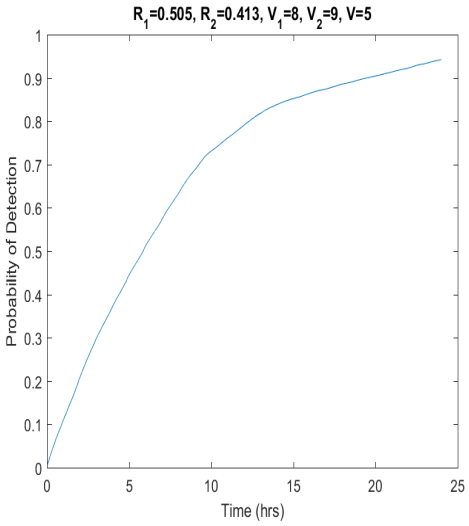


Figure A.334. P_D vs Time

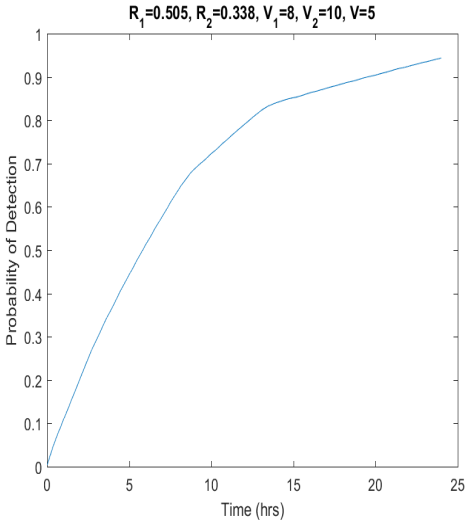


Figure A.335. P_D vs Time

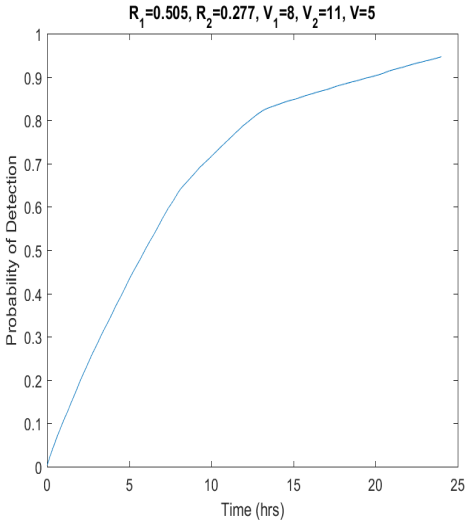


Figure A.336. P_D vs Time

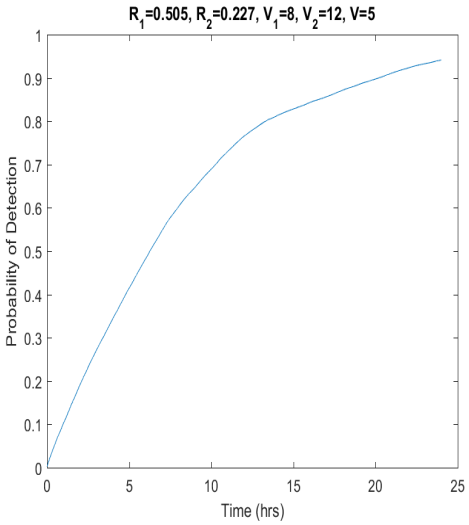


Figure A.337. P_D vs Time

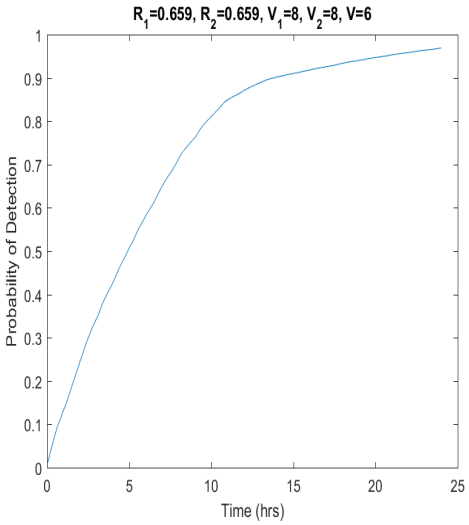


Figure A.338. P_D vs Time

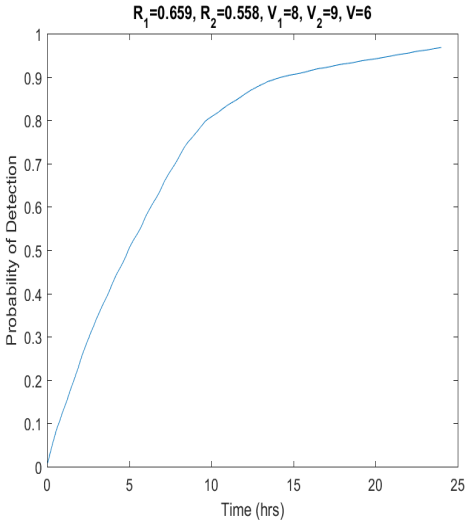


Figure A.339. P_D vs Time

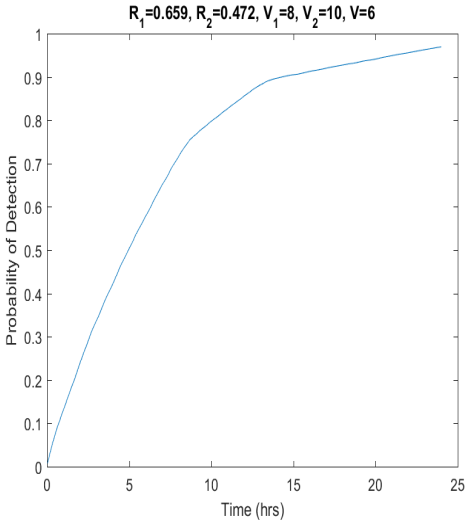


Figure A.340. P_D vs Time

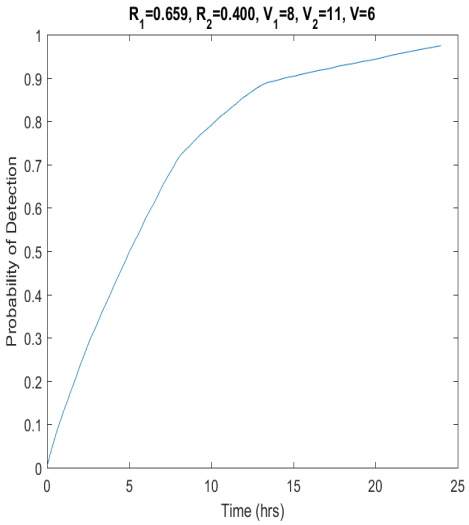


Figure A.341. P_D vs Time

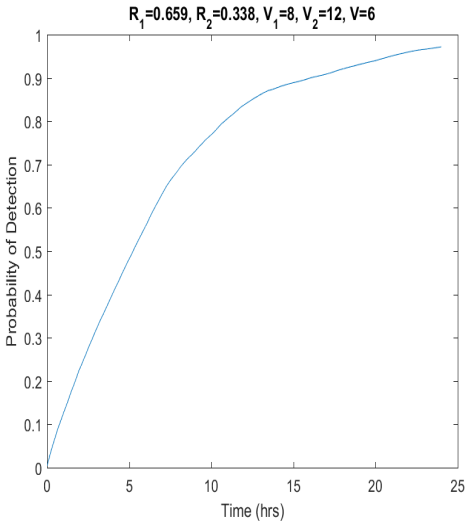


Figure A.342. P_D vs Time

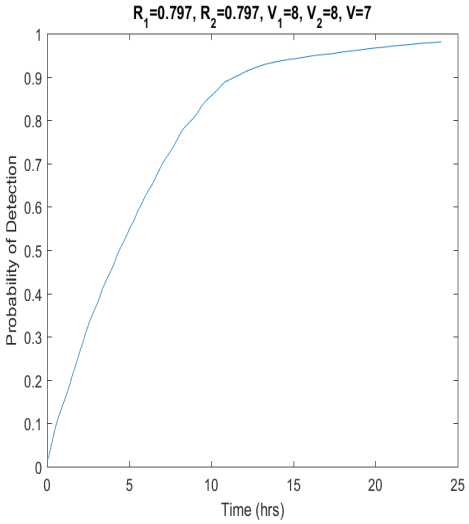


Figure A.343. P_D vs Time

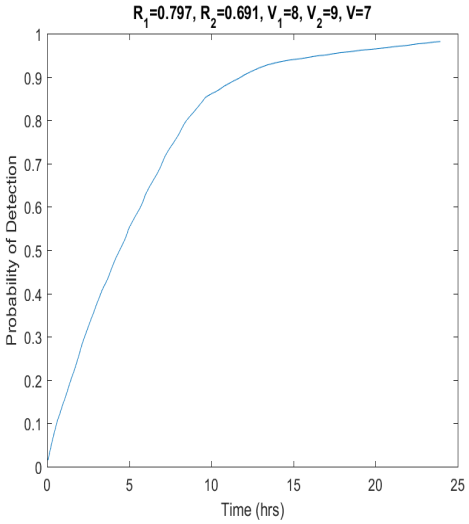


Figure A.344. P_D vs Time

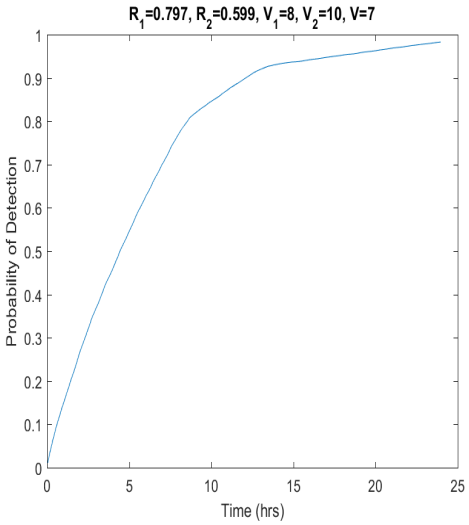


Figure A.345. P_D vs Time

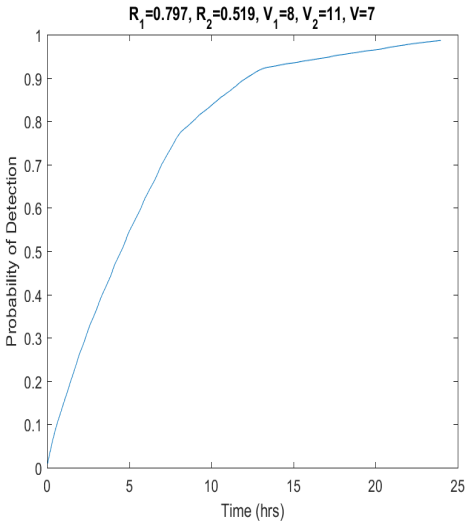


Figure A.346. P_D vs Time

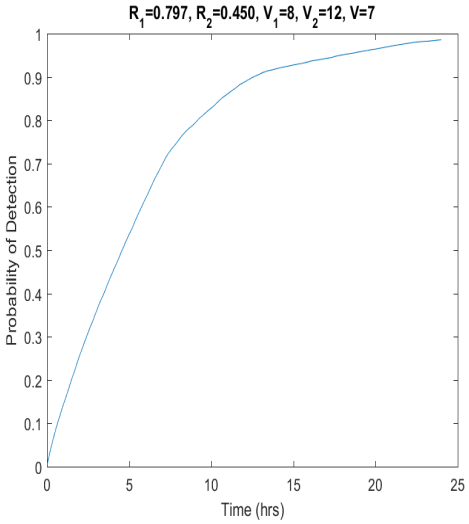


Figure A.347. P_D vs Time

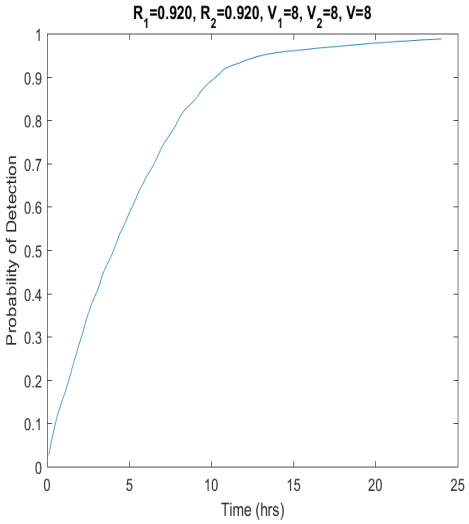


Figure A.348. P_D vs Time

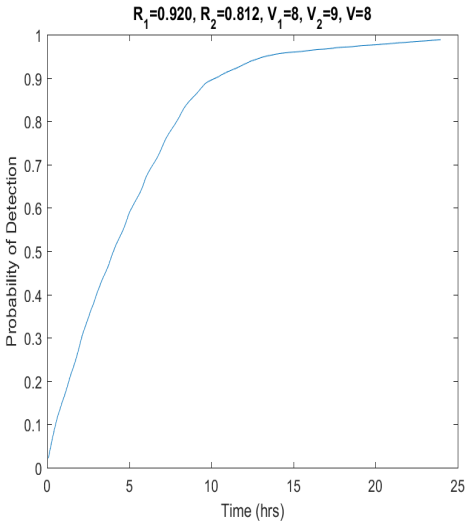


Figure A.349. P_D vs Time

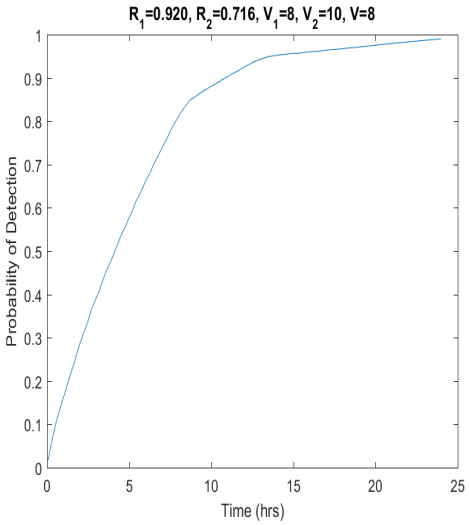


Figure A.350. P_D vs Time

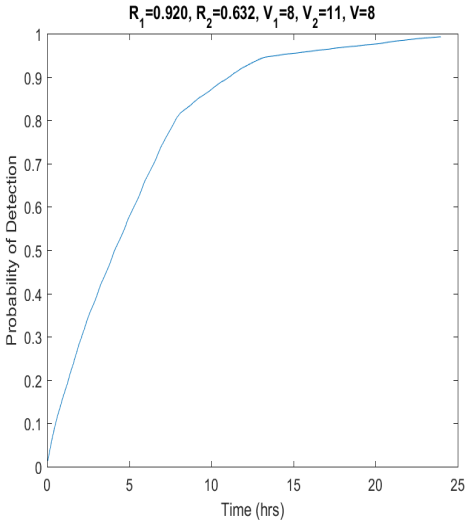


Figure A.351. P_D vs Time

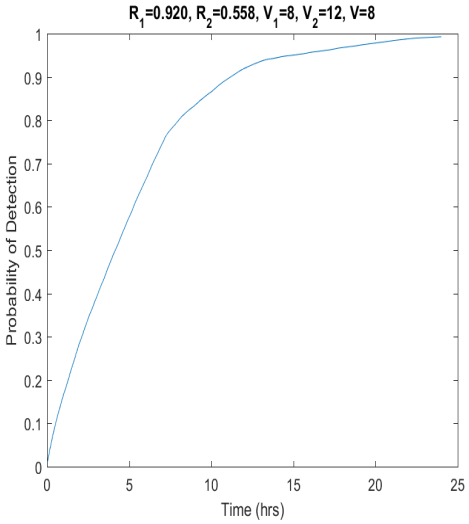


Figure A.352. P_D vs Time

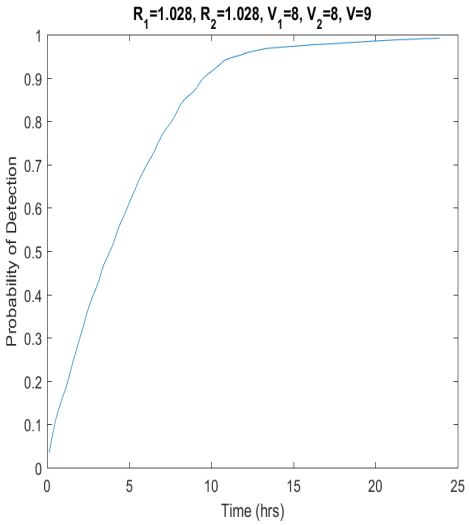


Figure A.353. P_D vs Time

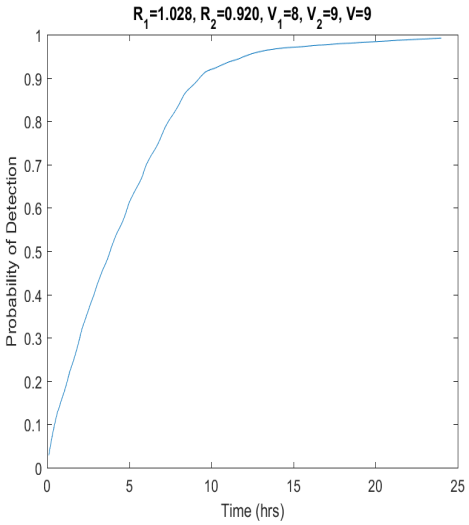


Figure A.354. P_D vs Time

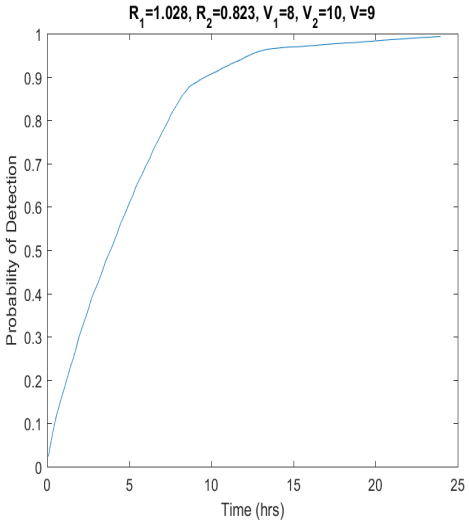


Figure A.355. P_D vs Time

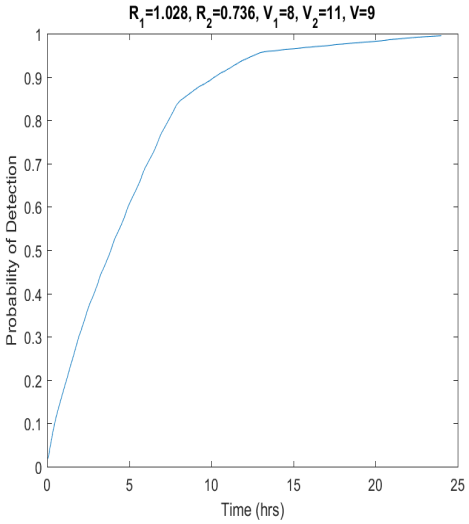


Figure A.356. P_D vs Time

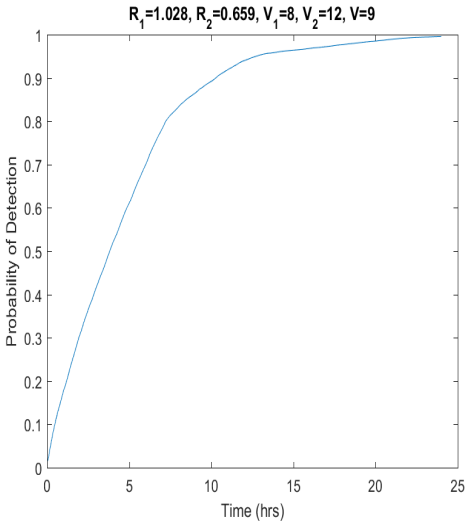


Figure A.357. P_D vs Time

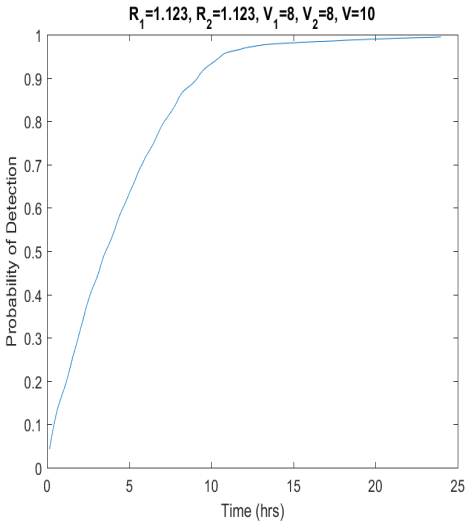


Figure A.358. P_D vs Time

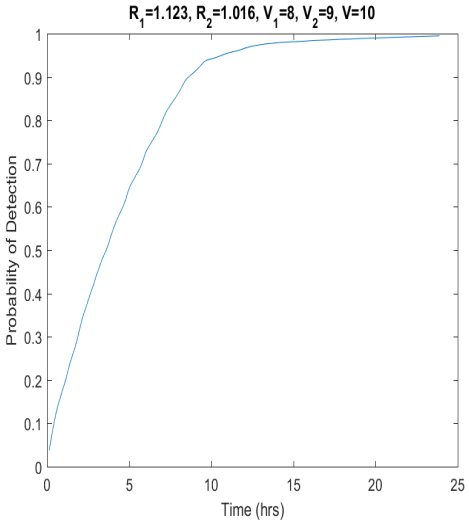


Figure A.359. P_D vs Time

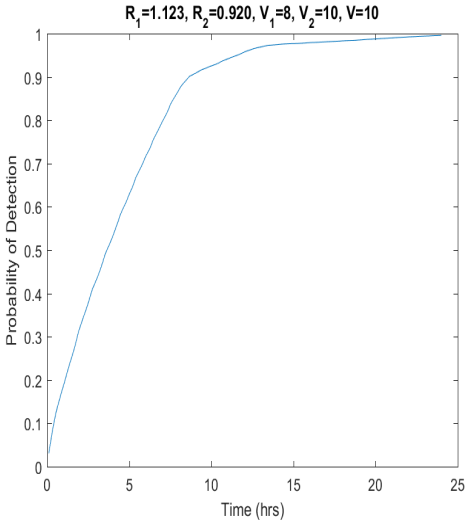


Figure A.360. P_D vs Time

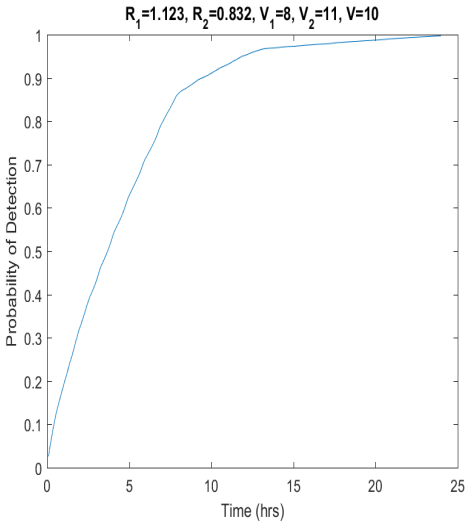


Figure A.361. P_D vs Time

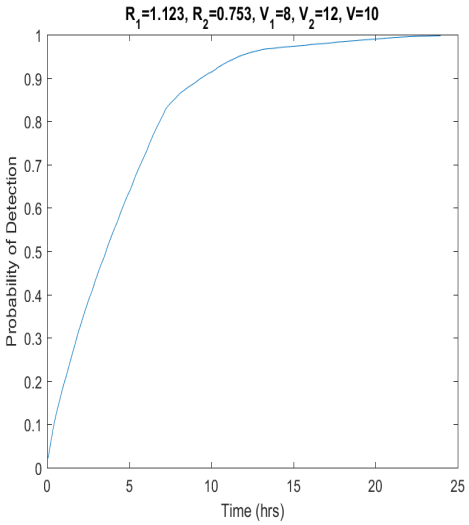


Figure A.362. P_D vs Time

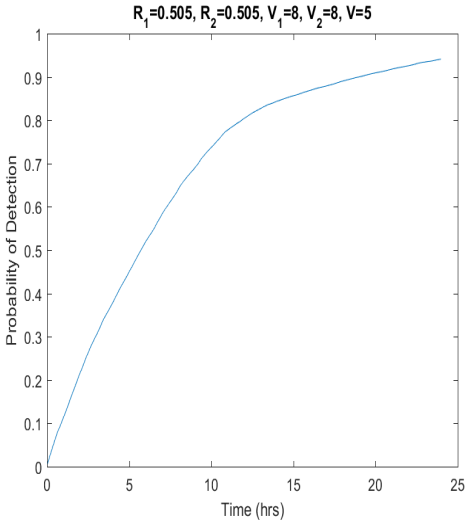


Figure A.363. P_D vs Time

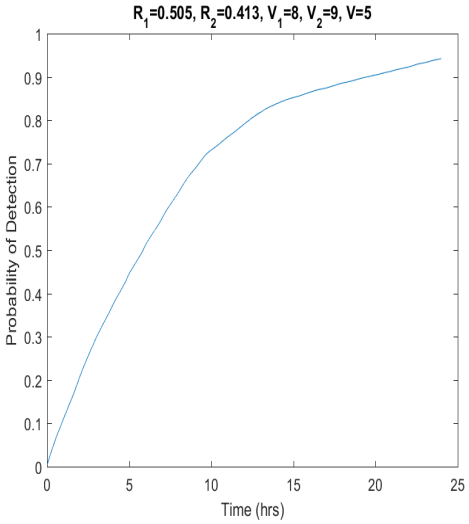


Figure A.364. P_D vs Time

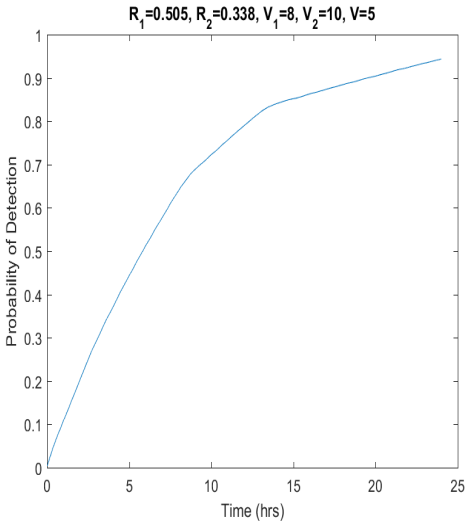


Figure A.365. P_D vs Time

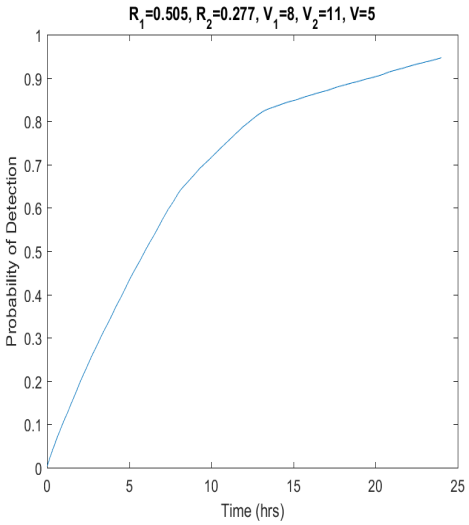


Figure A.366. P_D vs Time

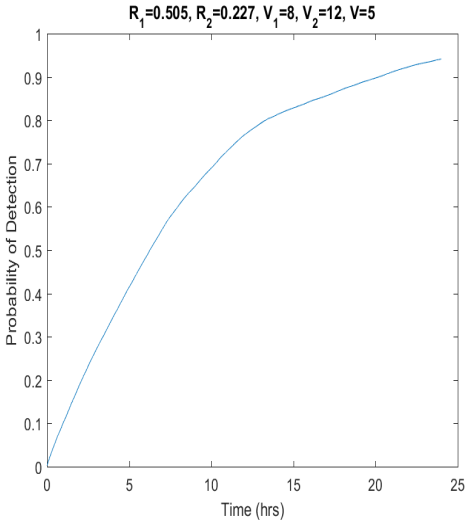


Figure A.367. P_D vs Time

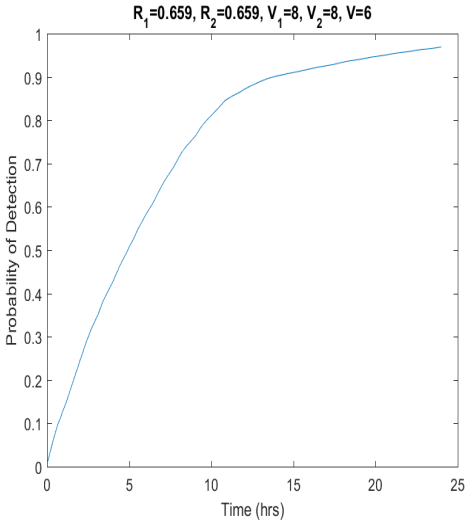


Figure A.368. P_D vs Time

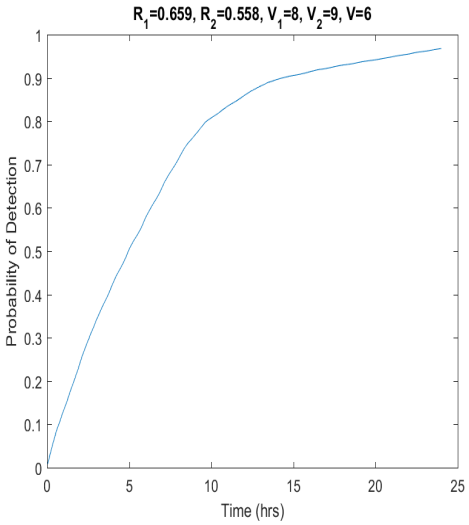


Figure A.369. P_D vs Time

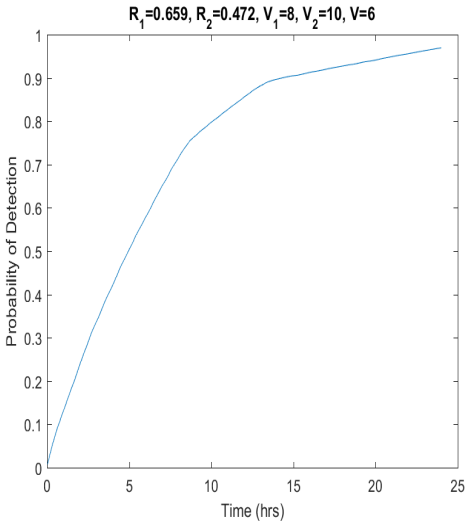


Figure A.370. P_D vs Time

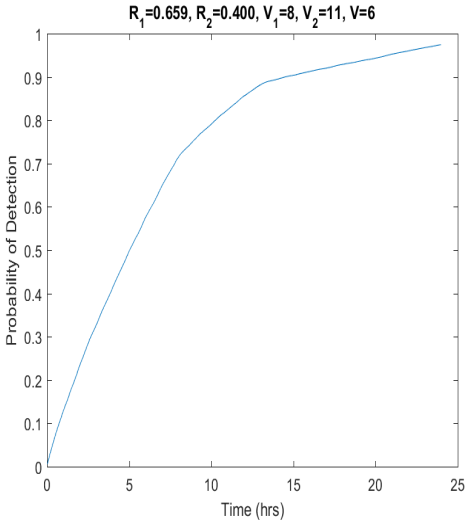


Figure A.371. P_D vs Time

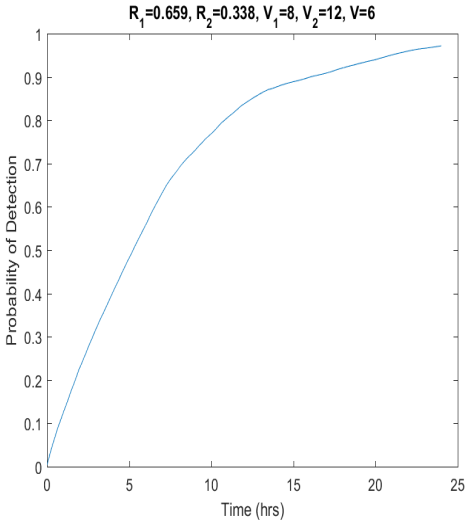


Figure A.372. P_D vs Time

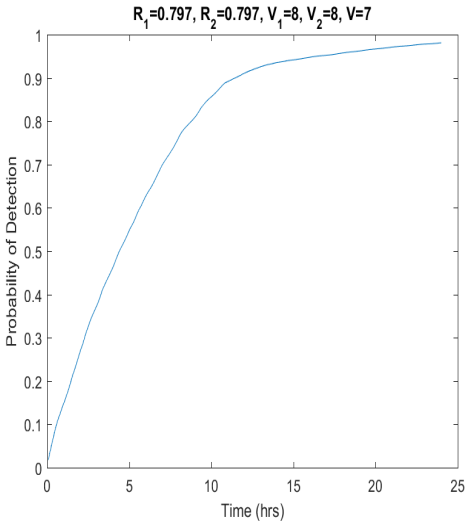


Figure A.373. P_D vs Time

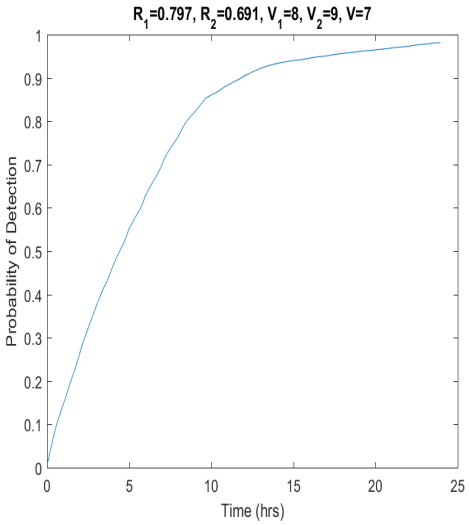


Figure A.374. P_D vs Time

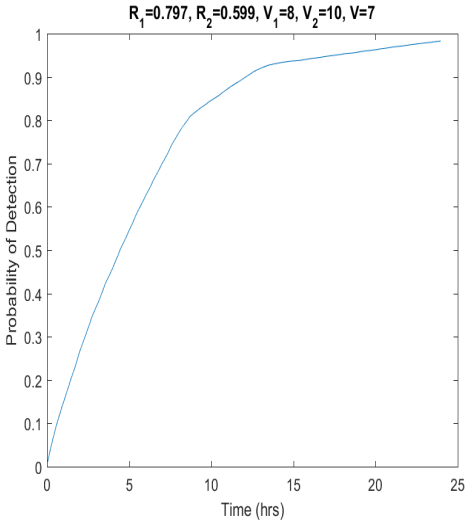


Figure A.375. P_D vs Time

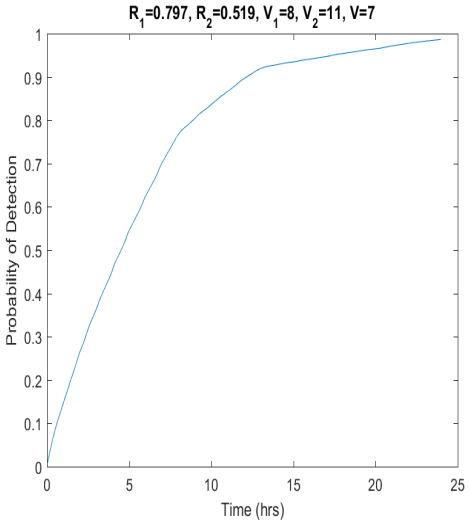


Figure A.376. P_D vs Time

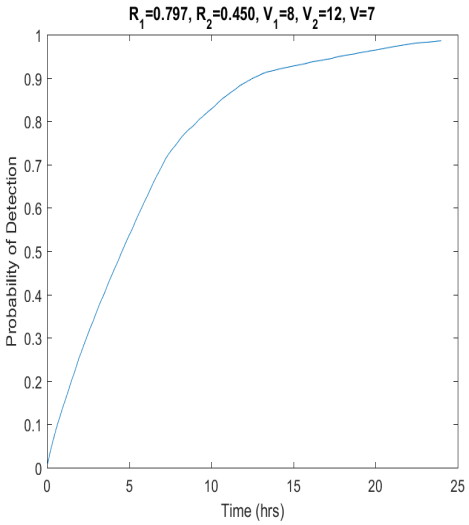


Figure A.377. P_D vs Time

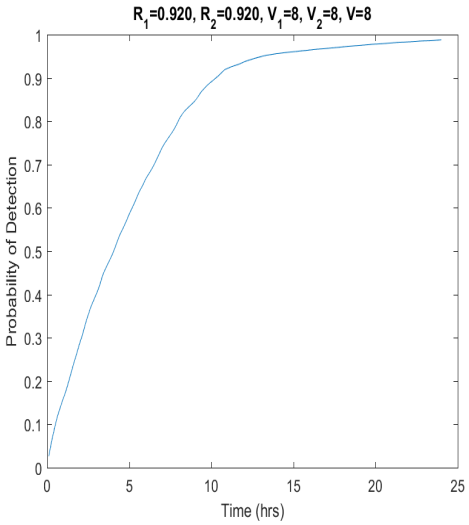


Figure A.378. P_D vs Time

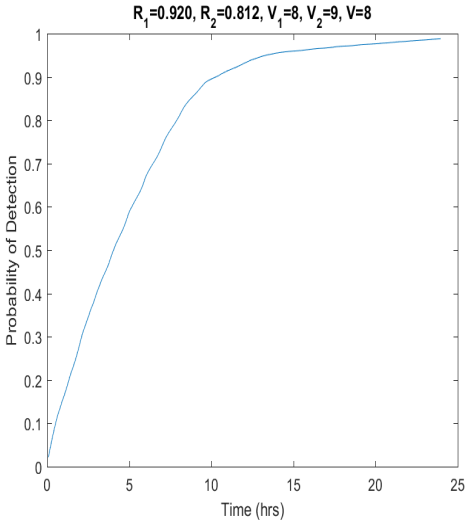


Figure A.379. P_D vs Time

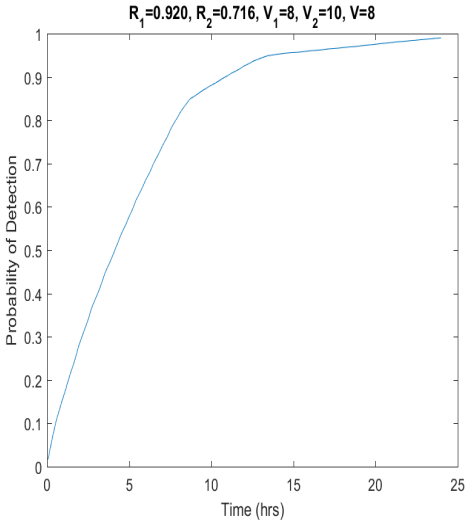


Figure A.380. P_D vs Time

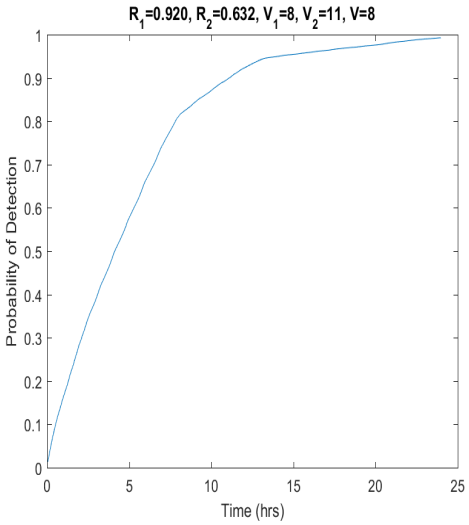


Figure A.381. P_D vs Time

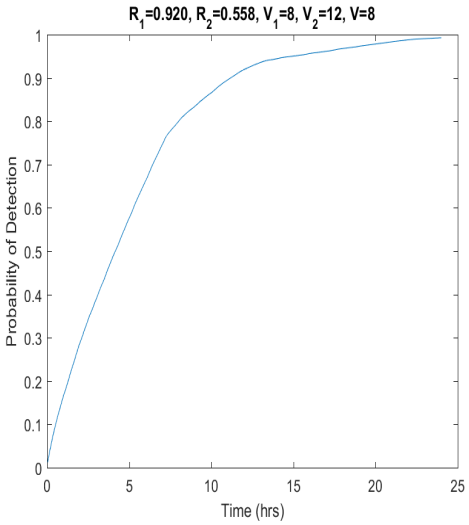


Figure A.382. P_D vs Time

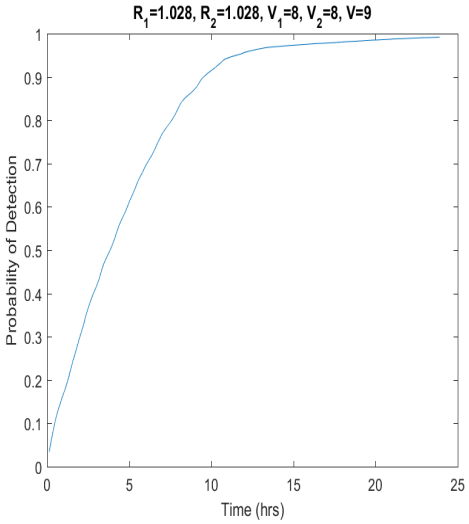


Figure A.383. P_D vs Time

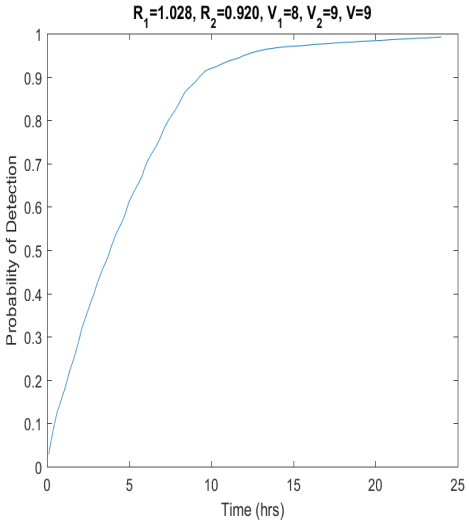


Figure A.384. P_D vs Time

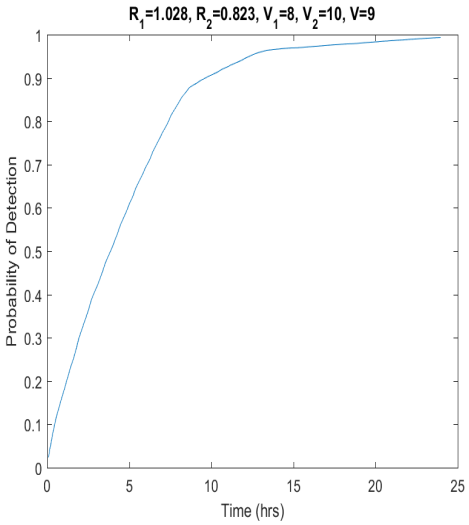


Figure A.385. P_D vs Time

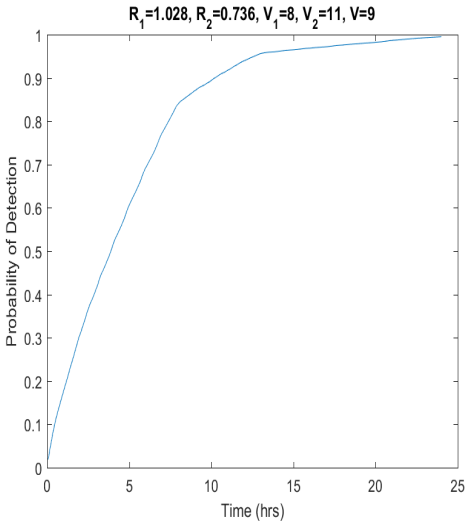


Figure A.386. P_D vs Time

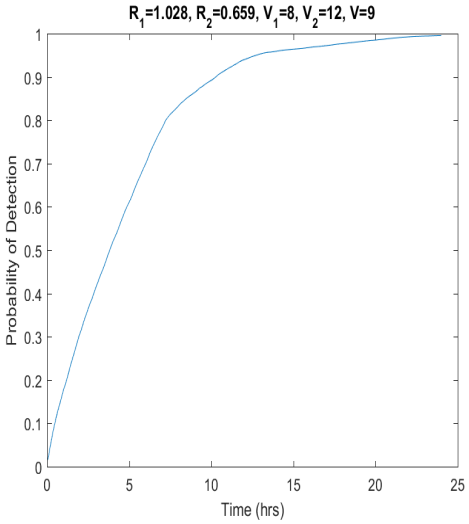


Figure A.387. P_D vs Time

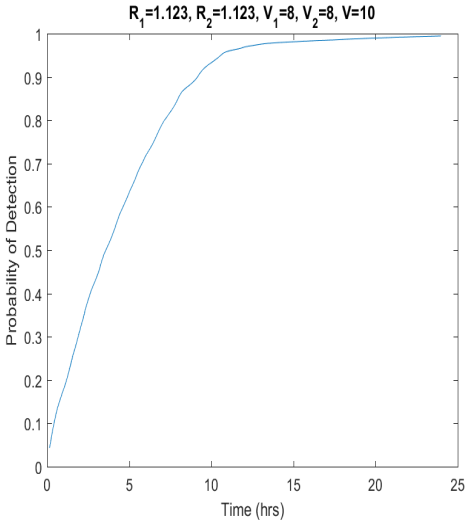


Figure A.388. P_D vs Time

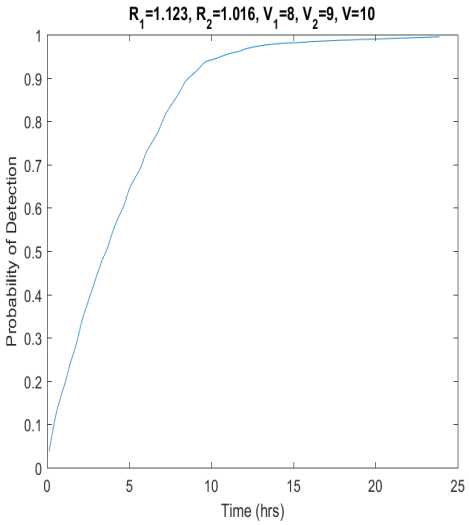


Figure A.389. P_D vs Time

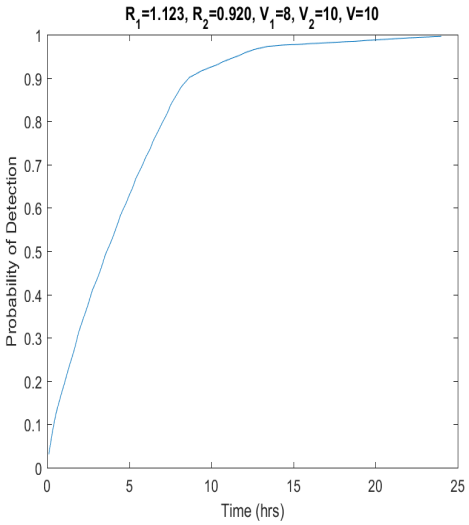


Figure A.390. P_D vs Time

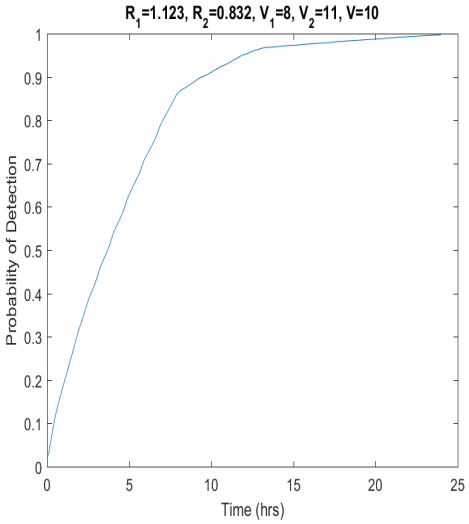


Figure A.391. P_D vs Time

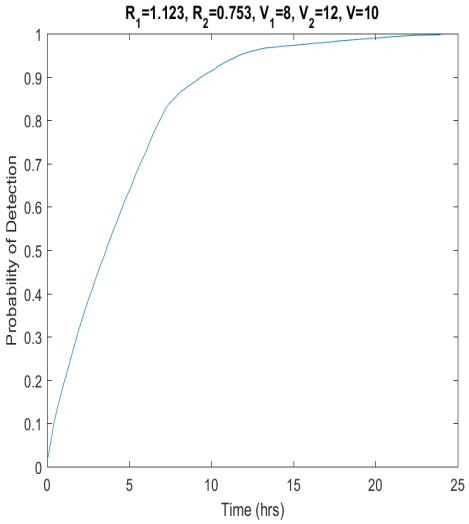


Figure A.392. P_D vs Time

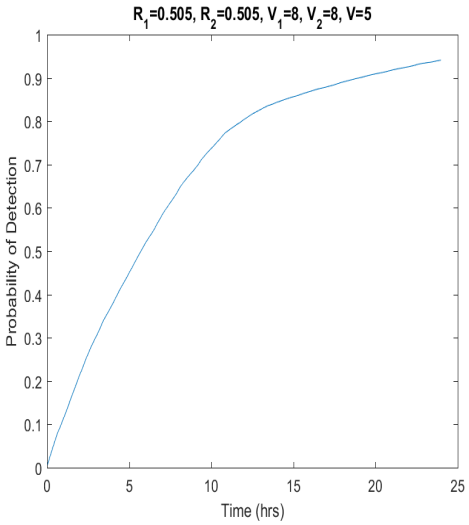


Figure A.393. P_D vs Time

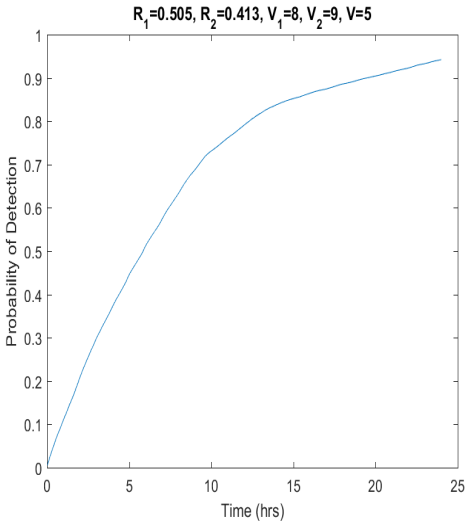


Figure A.394. P_D vs Time

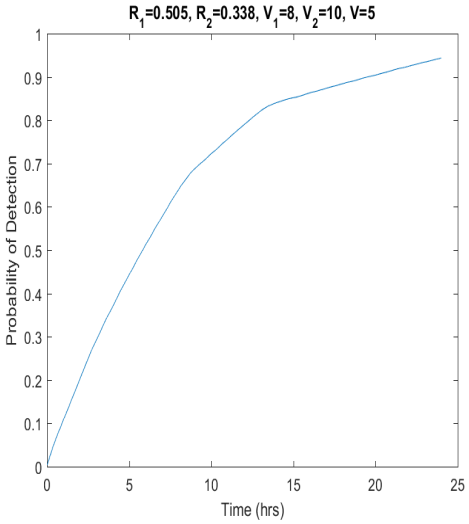


Figure A.395. P_D vs Time

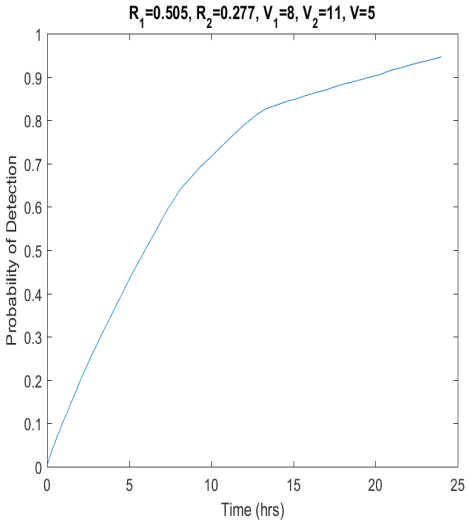


Figure A.396. P_D vs Time

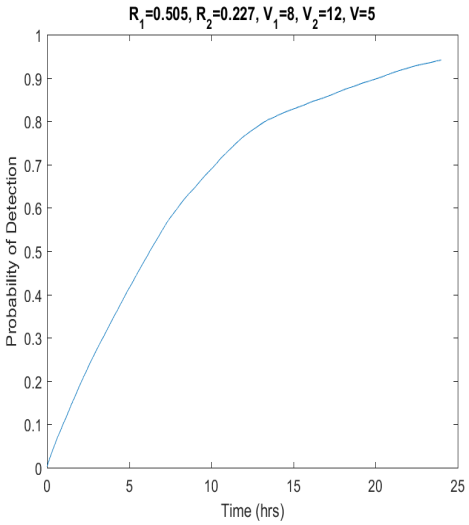


Figure A.397. P_D vs Time

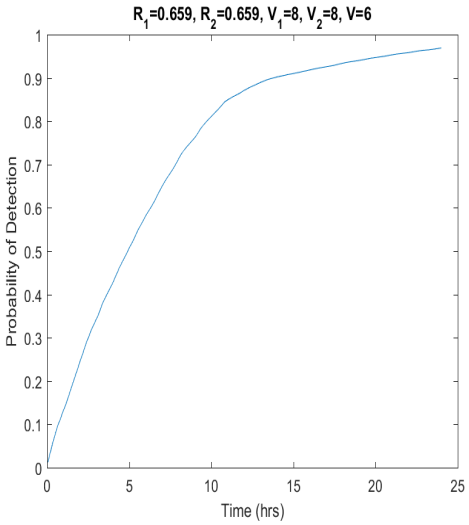


Figure A.398. P_D vs Time

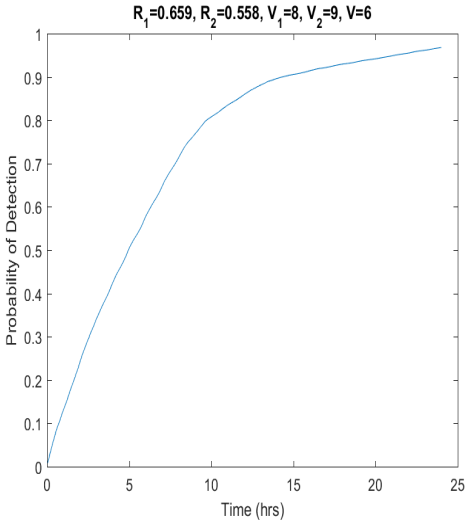


Figure A.399. P_D vs Time

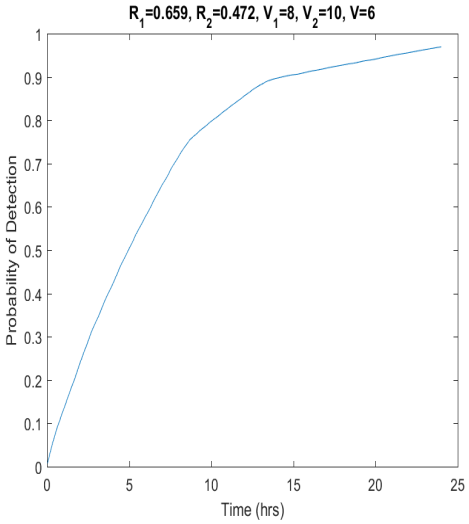


Figure A.400. P_D vs Time

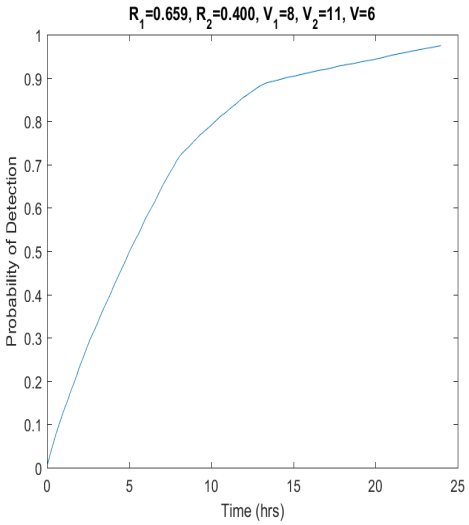


Figure A.401. P_D vs Time

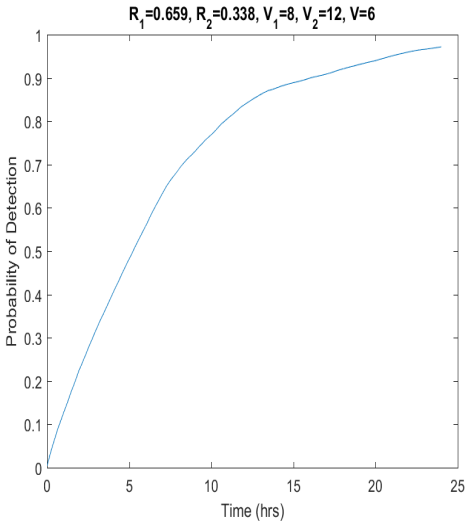


Figure A.402. P_D vs Time

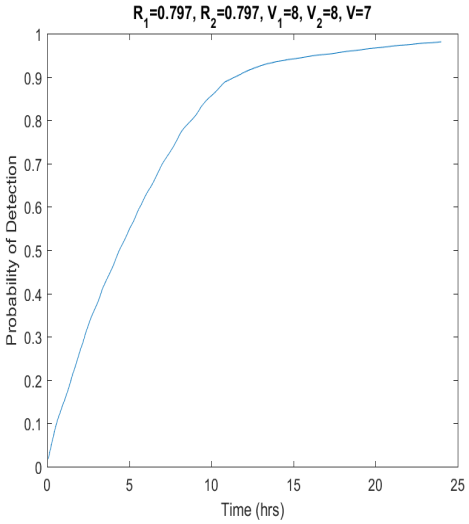


Figure A.403. P_D vs Time

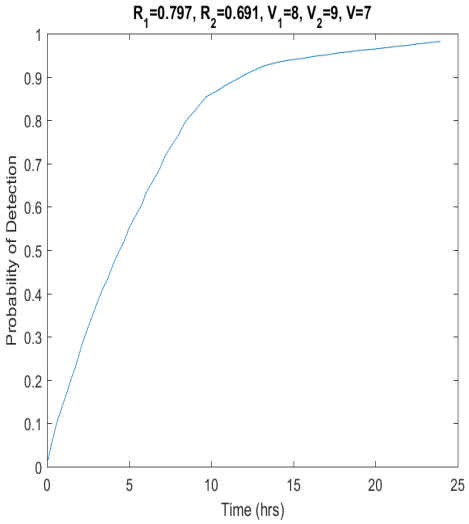


Figure A.404. P_D vs Time

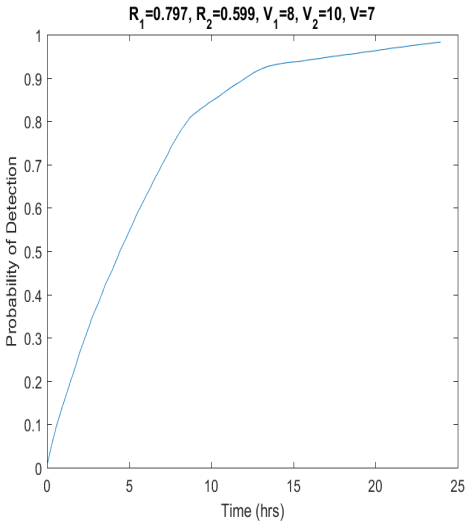


Figure A.405. P_D vs Time

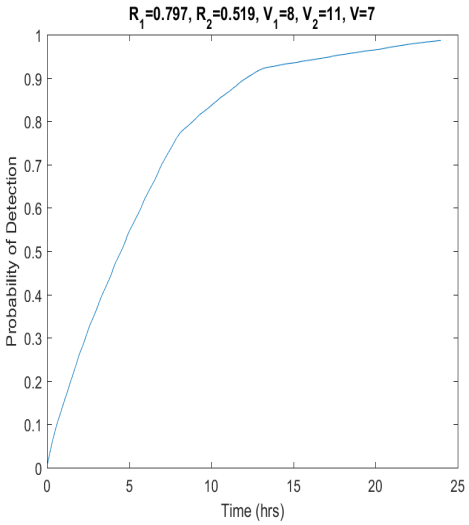


Figure A.406. P_D vs Time

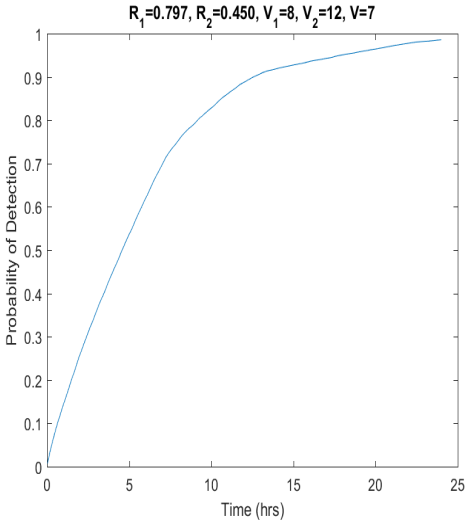


Figure A.407. P_D vs Time

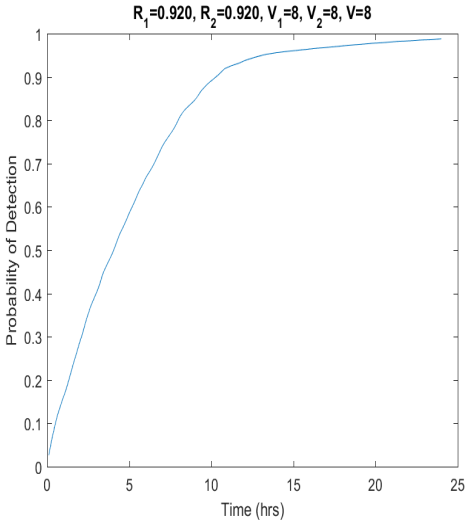


Figure A.408. P_D vs Time

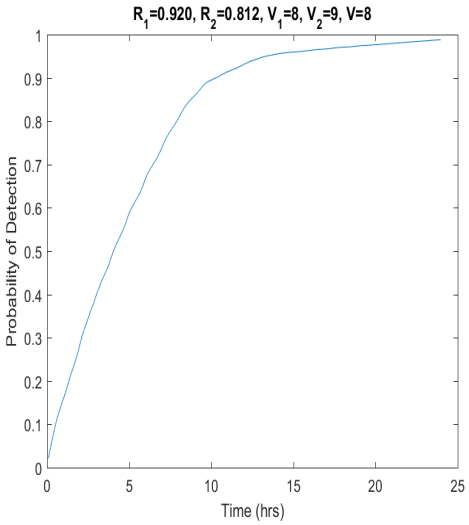


Figure A.409. P_D vs Time

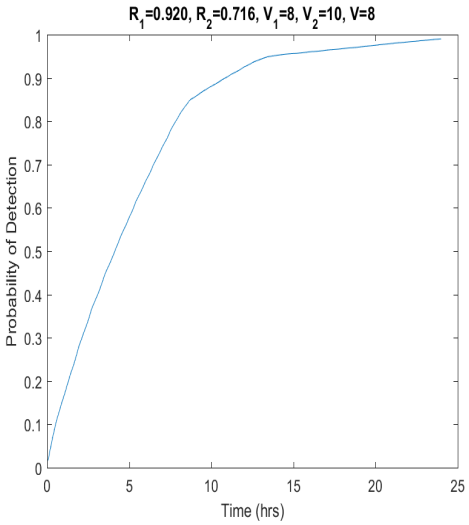


Figure A.410. P_D vs Time

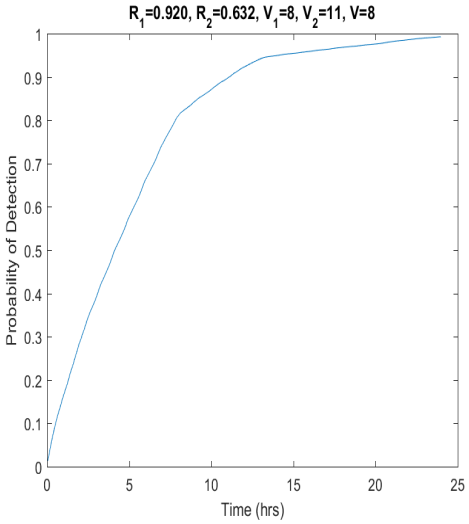


Figure A.411. P_D vs Time

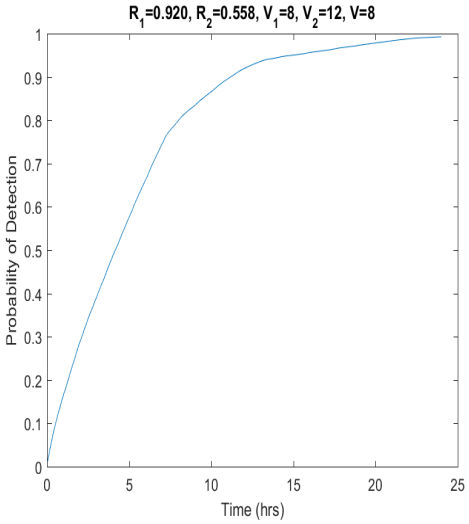


Figure A.412. P_D vs Time

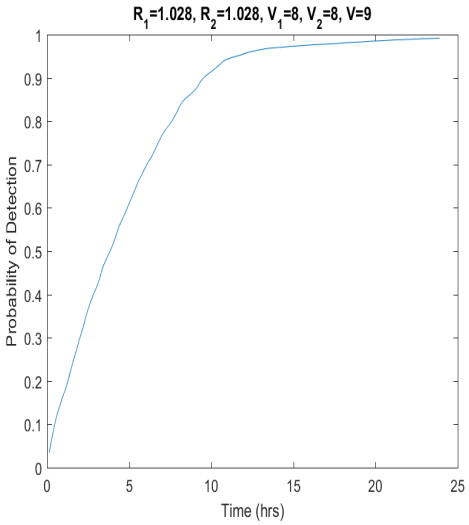


Figure A.413. P_D vs Time

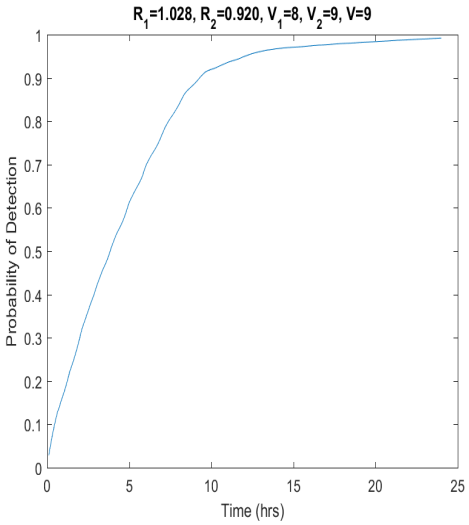


Figure A.414. P_D vs Time

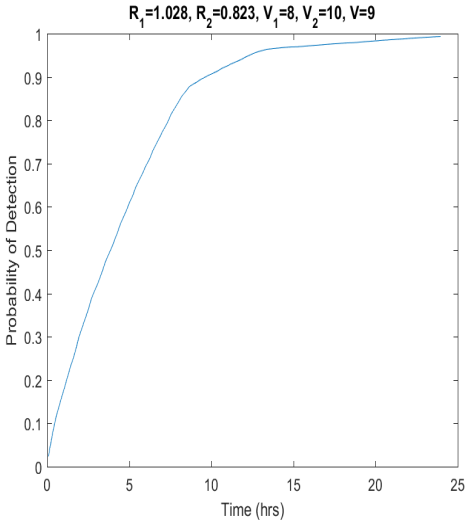


Figure A.415. P_D vs Time

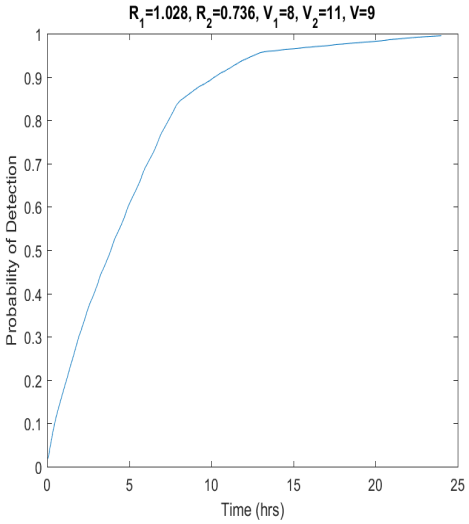


Figure A.416. P_D vs Time

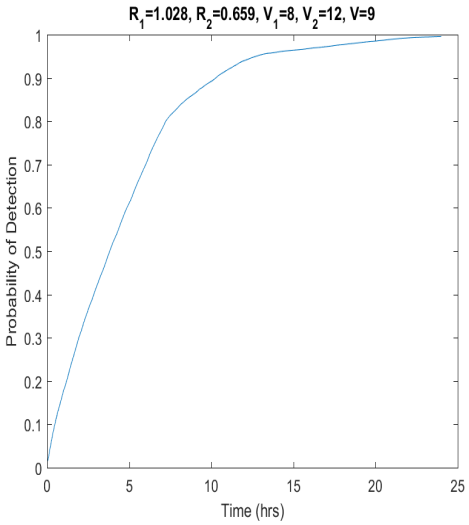


Figure A.417. P_D vs Time

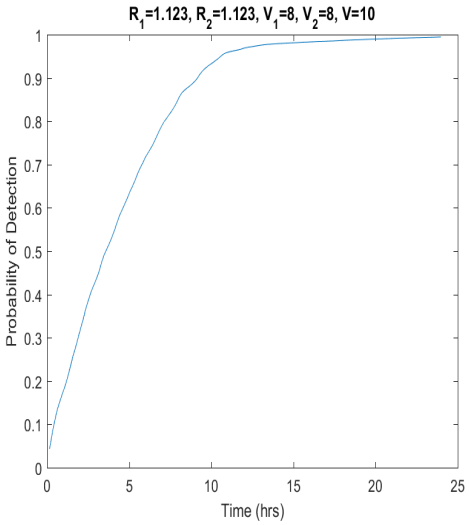


Figure A.418. P_D vs Time

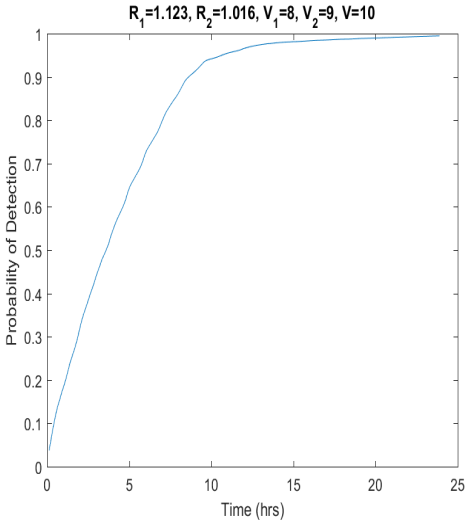


Figure A.419. P_D vs Time

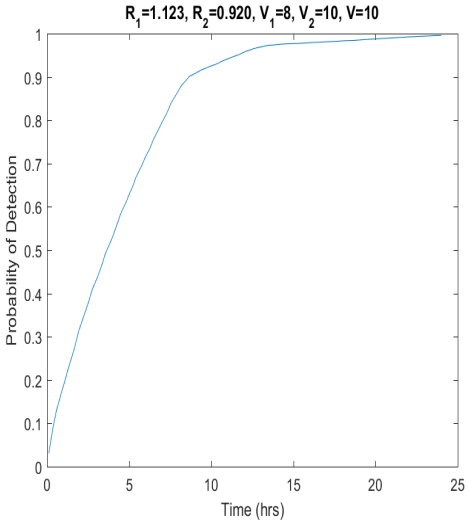


Figure A.420. P_D vs Time

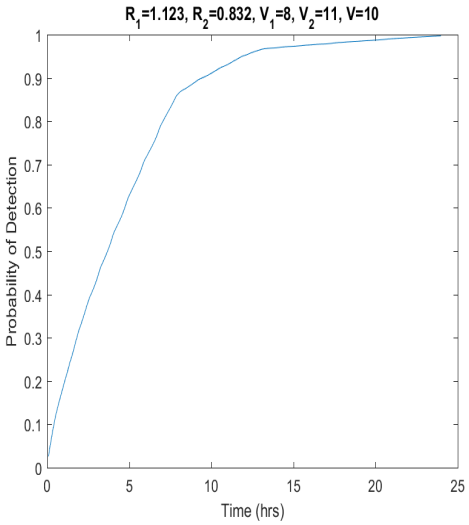


Figure A.421. P_D vs Time

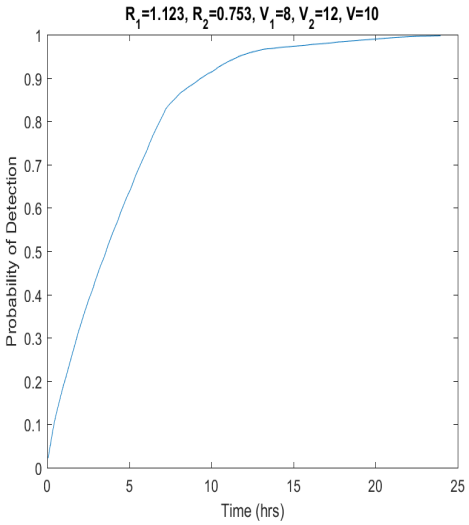


Figure A.422. P_D vs Time

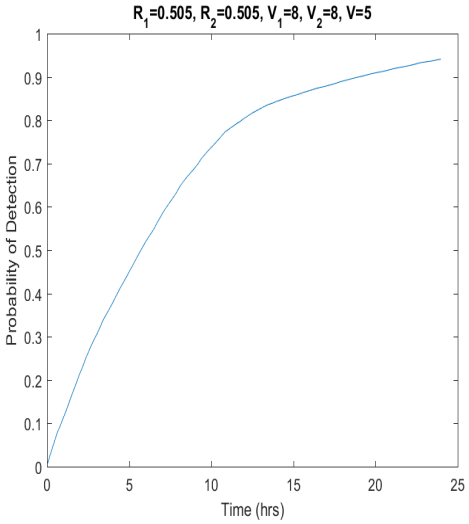


Figure A.423. P_D vs Time

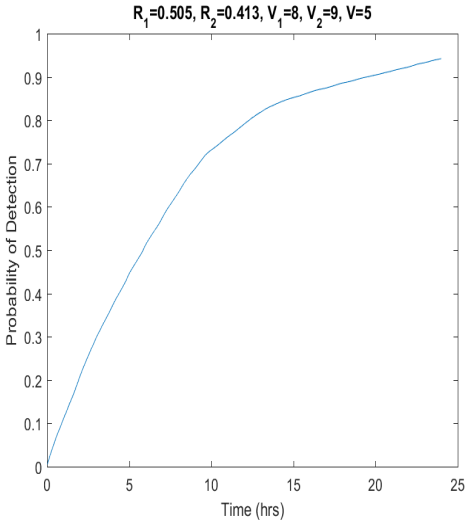


Figure A.424. P_D vs Time

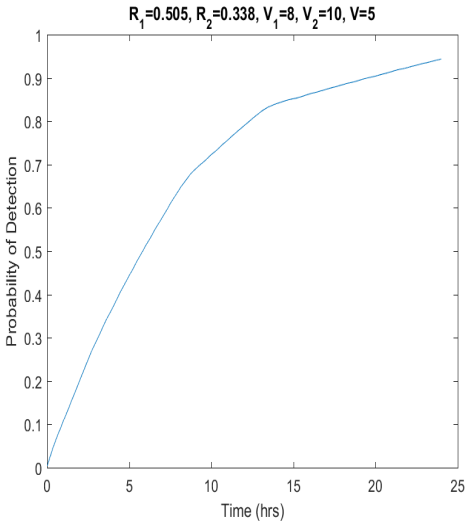


Figure A.425. P_D vs Time

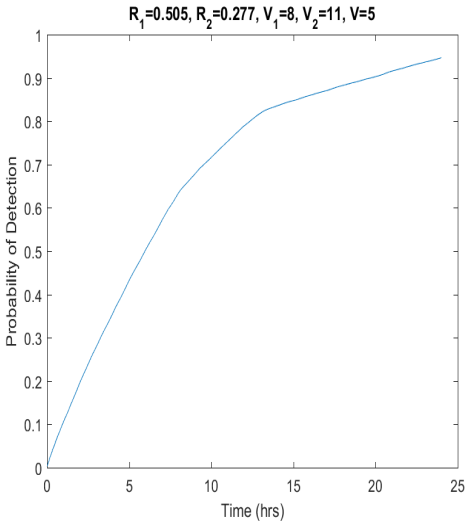


Figure A.426. P_D vs Time

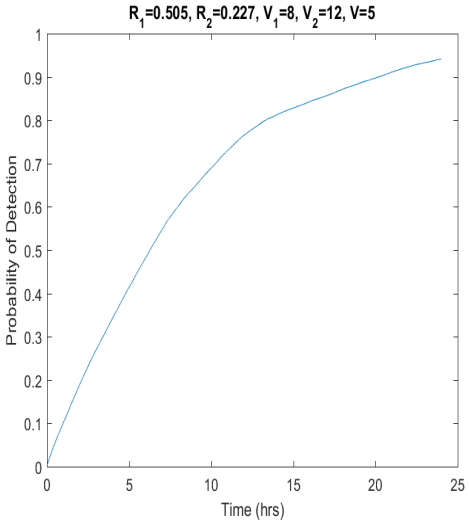


Figure A.427. P_D vs Time

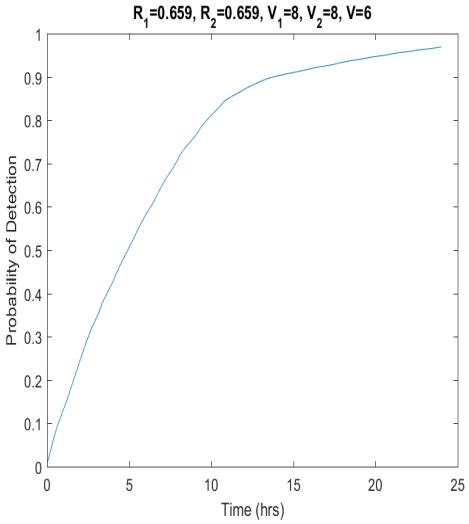


Figure A.428. P_D vs Time

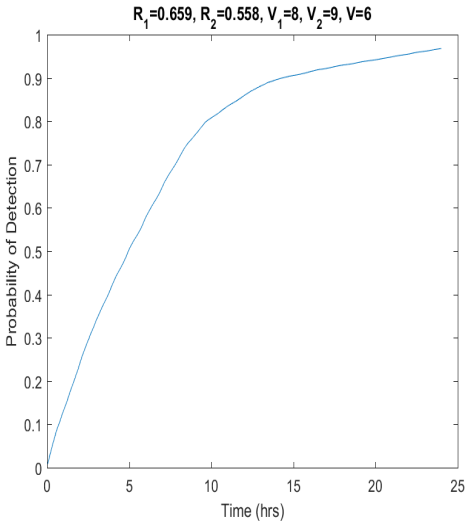


Figure A.429. P_D vs Time

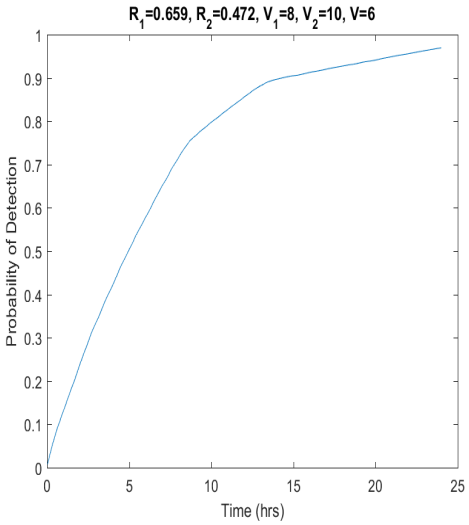


Figure A.430. P_D vs Time

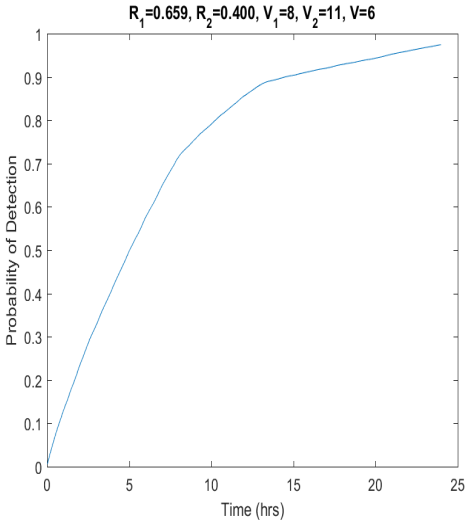


Figure A.431. P_D vs Time

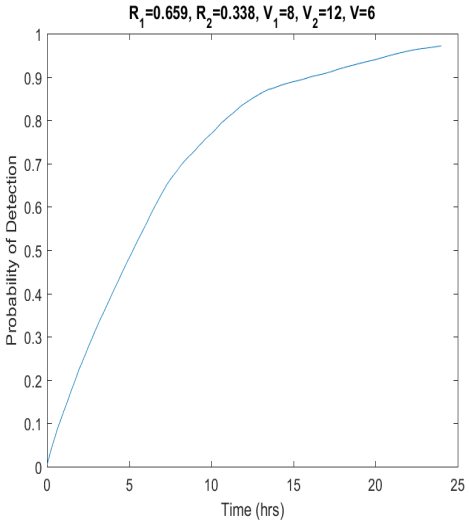


Figure A.432. P_D vs Time

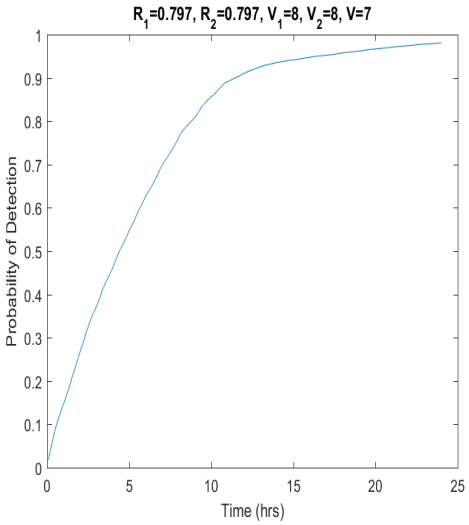


Figure A.433. P_D vs Time

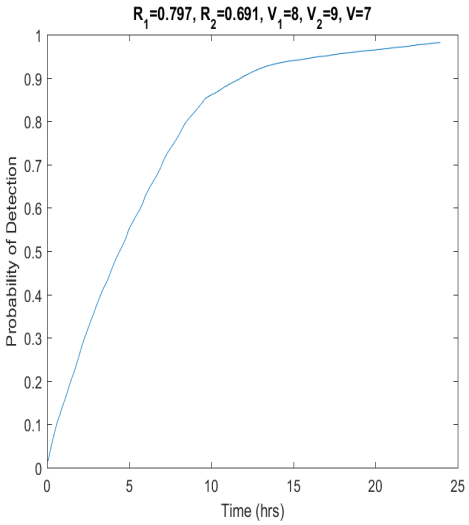


Figure A.434. P_D vs Time

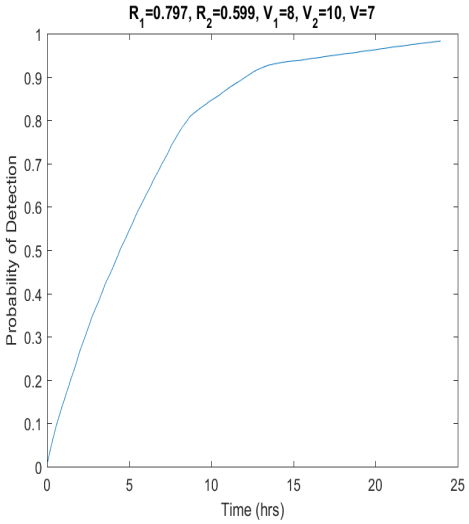


Figure A.435. P_D vs Time

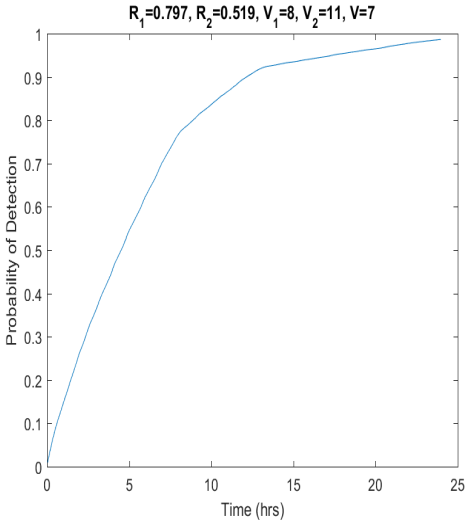


Figure A.436. P_D vs Time

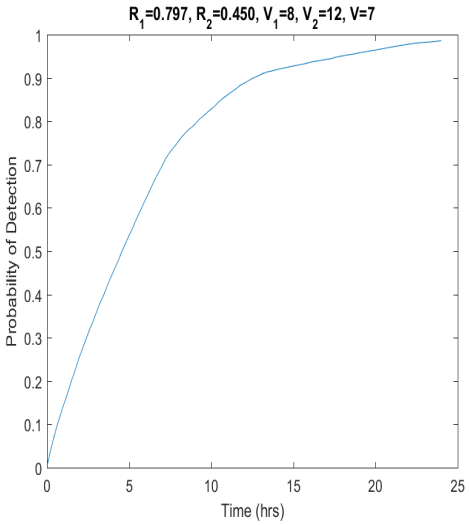


Figure A.437. P_D vs Time

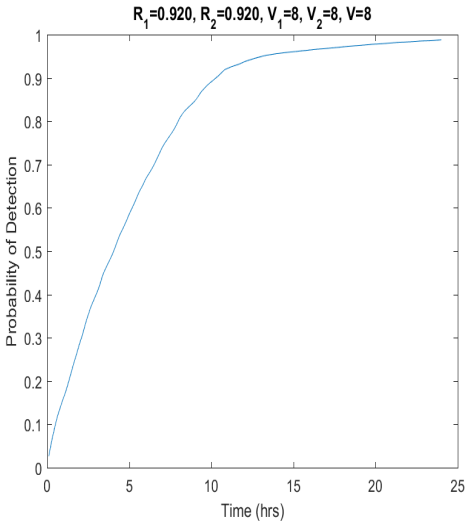


Figure A.438. P_D vs Time

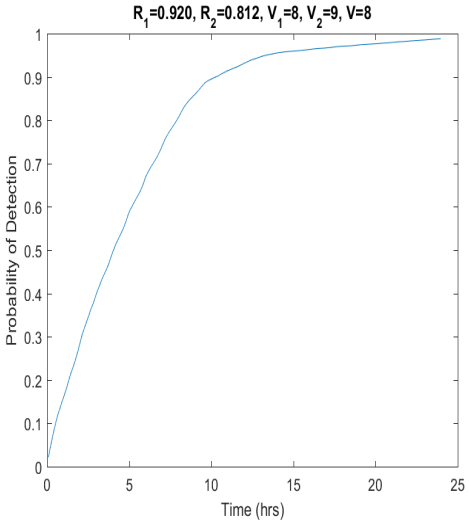


Figure A.439. P_D vs Time

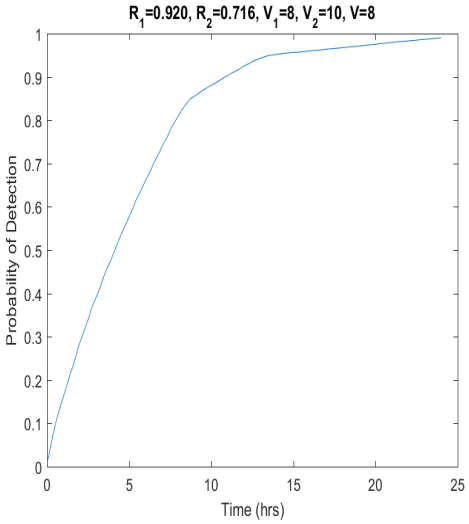


Figure A.440. P_D vs Time

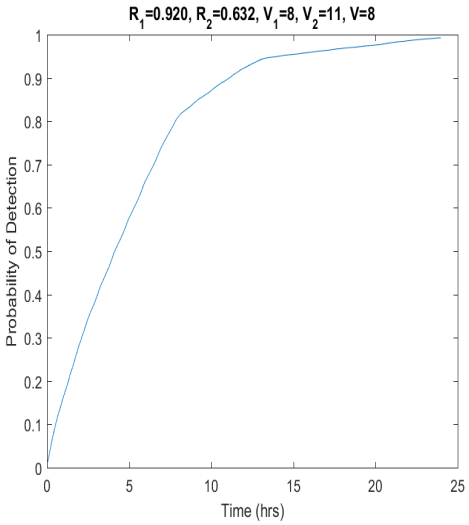


Figure A.441. P_D vs Time

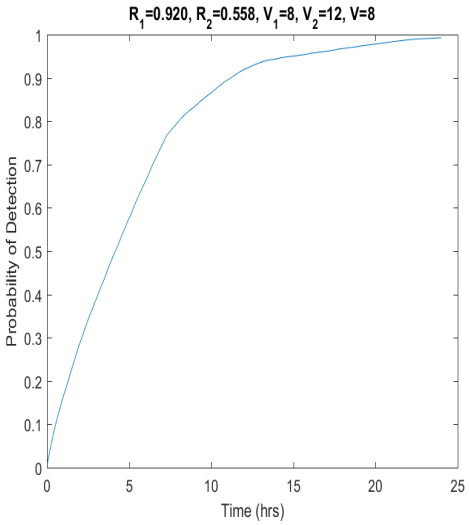


Figure A.442. P_D vs Time

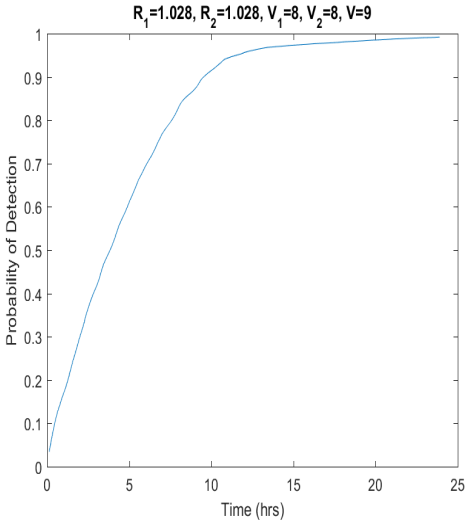


Figure A.443. P_D vs Time

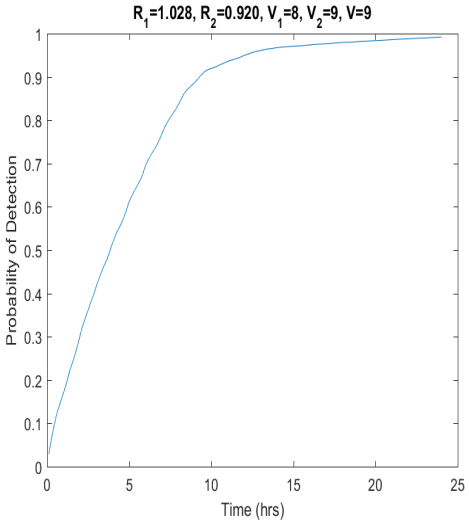


Figure A.444. P_D vs Time

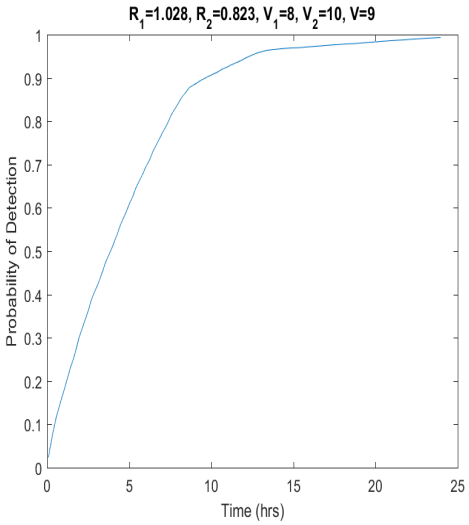


Figure A.445. P_D vs Time

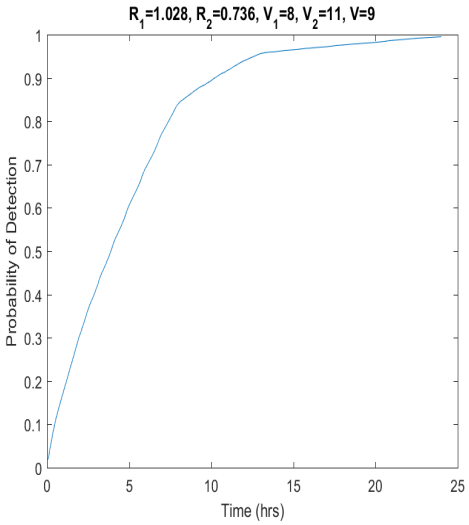


Figure A.446. P_D vs Time

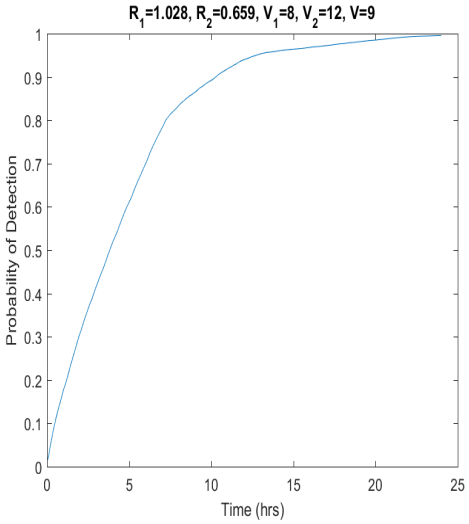


Figure A.447. P_D vs Time

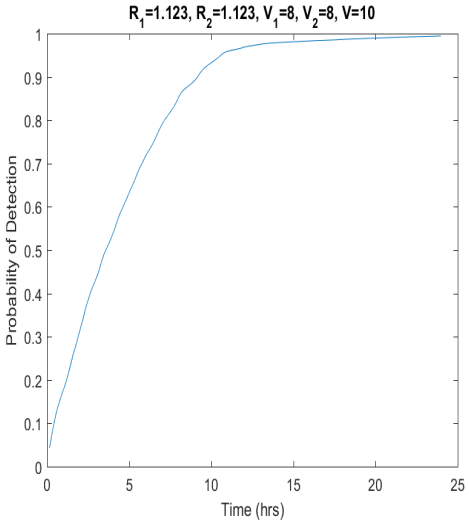


Figure A.448. P_D vs Time

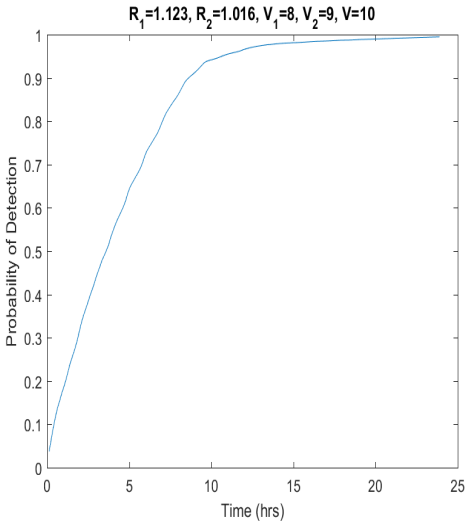


Figure A.449. P_D vs Time

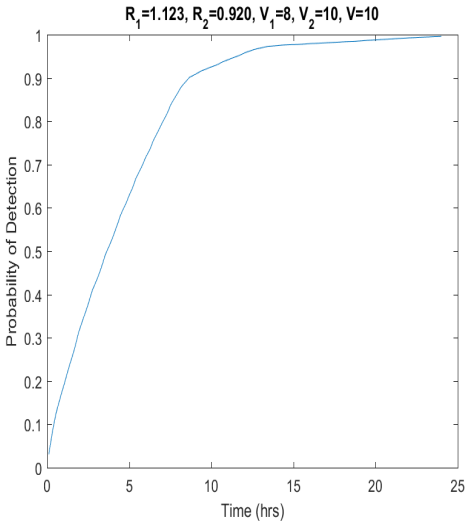


Figure A.450. P_D vs Time

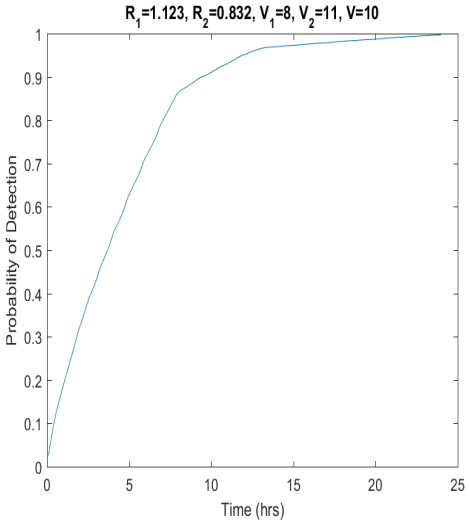


Figure A.451. P_D vs Time

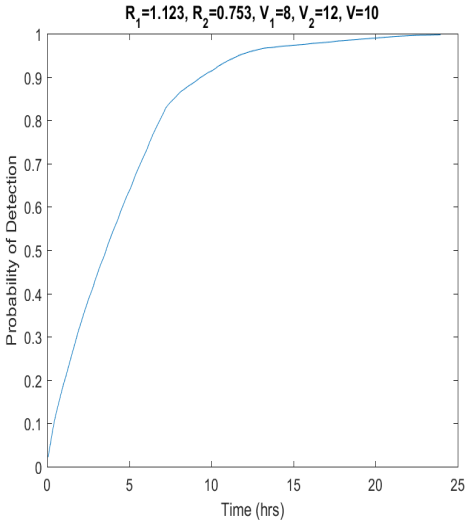


Figure A.452. P_D vs Time

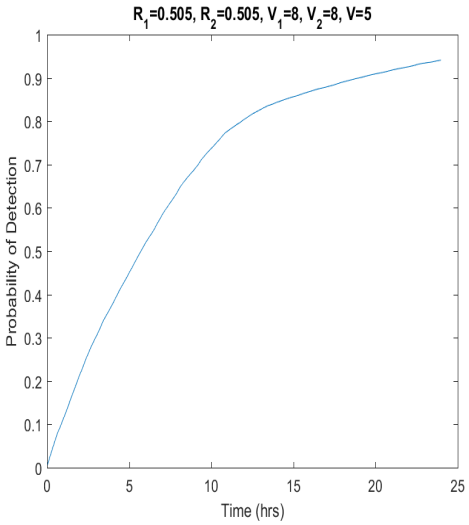


Figure A.453. P_D vs Time

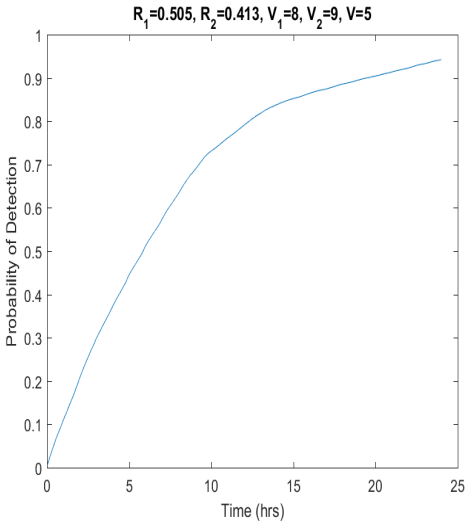


Figure A.454. P_D vs Time

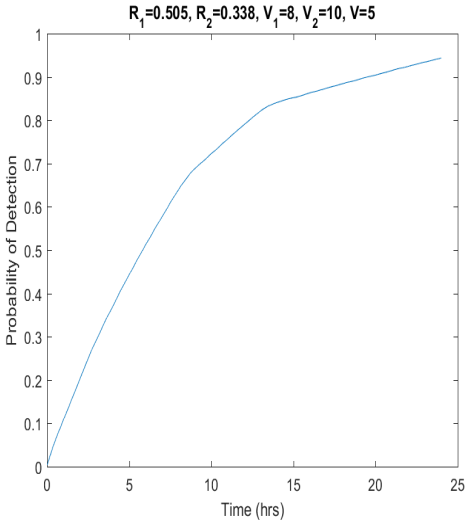


Figure A.455. P_D vs Time

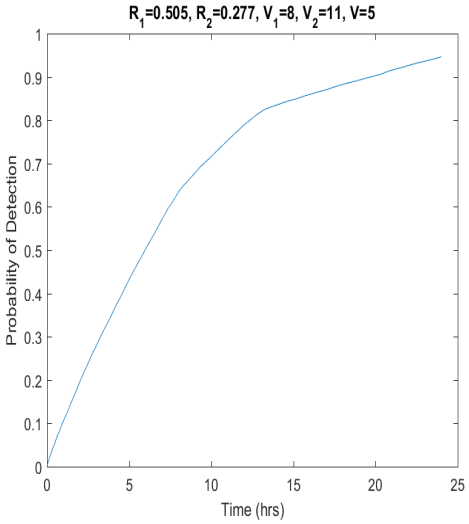


Figure A.456. P_D vs Time

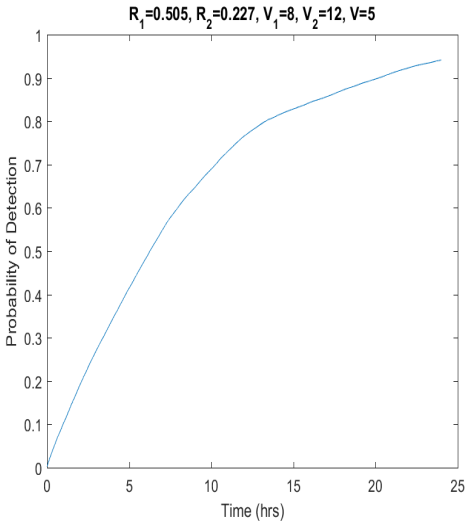


Figure A.457. P_D vs Time

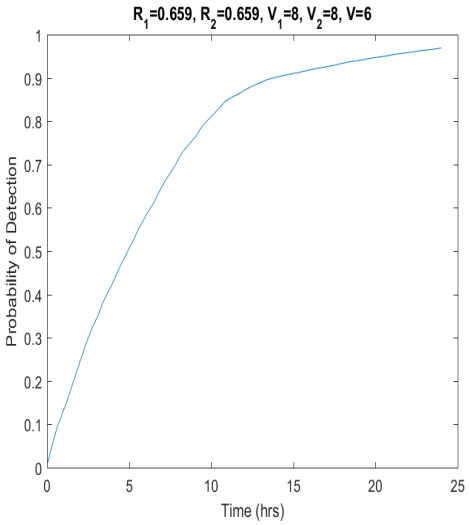


Figure A.458. P_D vs Time

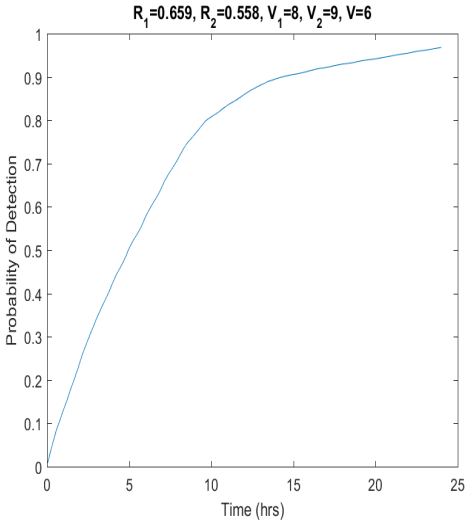


Figure A.459. P_D vs Time

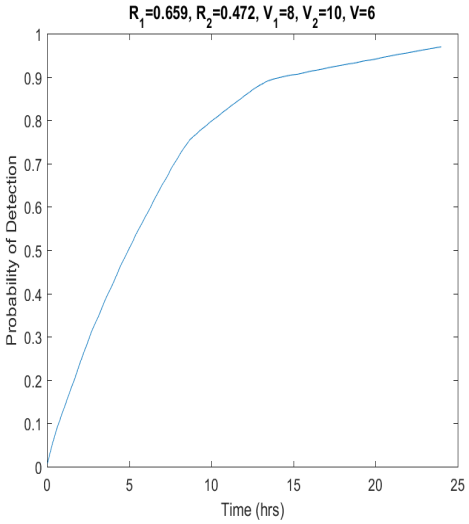


Figure A.460. P_D vs Time

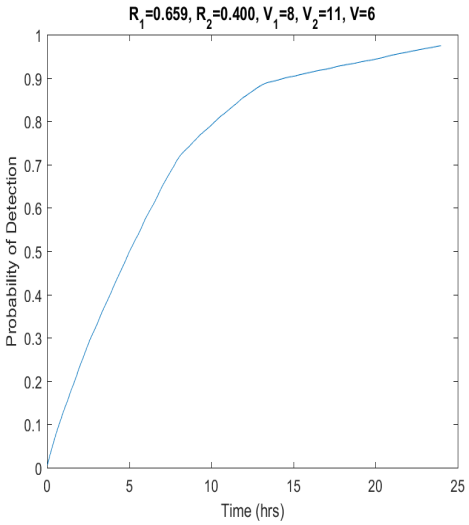


Figure A.461. P_D vs Time

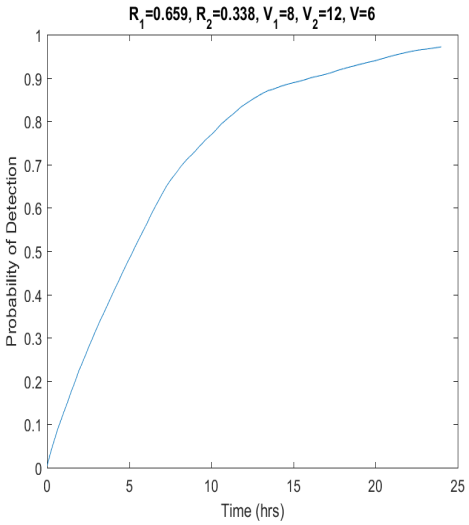


Figure A.462. P_D vs Time

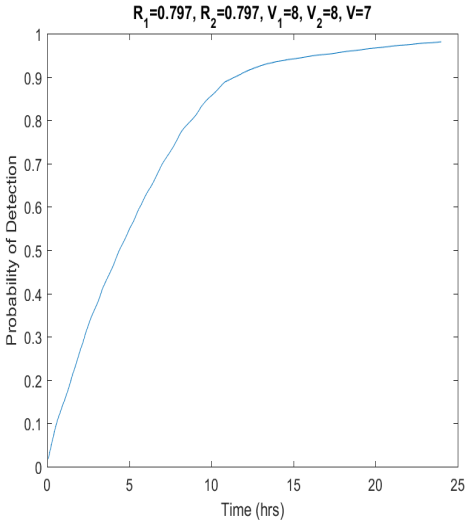


Figure A.463. P_D vs Time

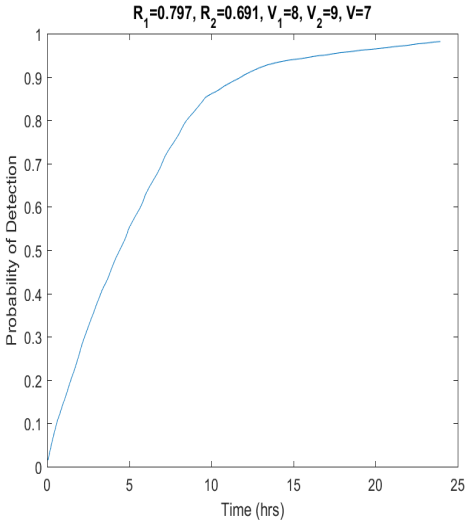


Figure A.464. P_D vs Time

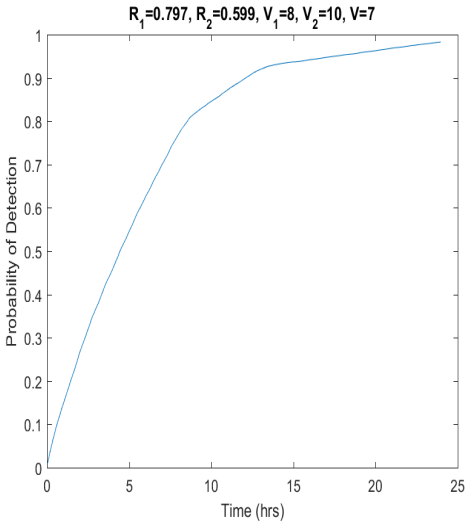


Figure A.465. P_D vs Time

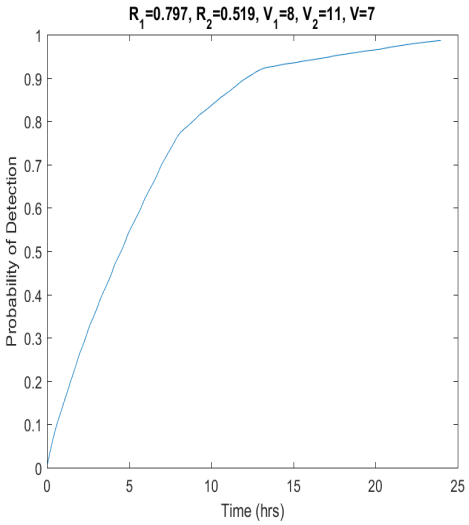


Figure A.466. P_D vs Time

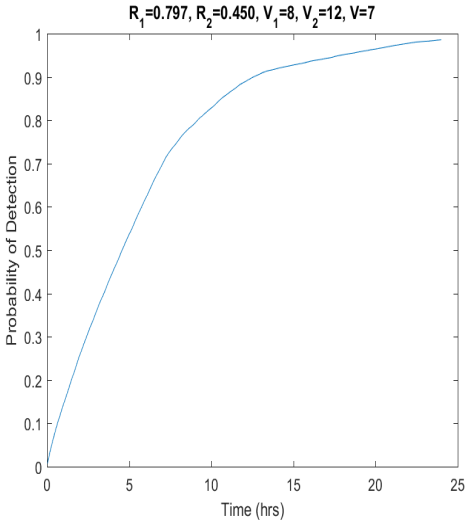


Figure A.467. P_D vs Time

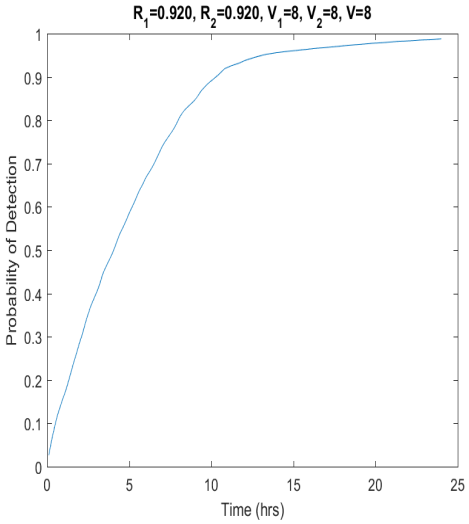


Figure A.468. P_D vs Time

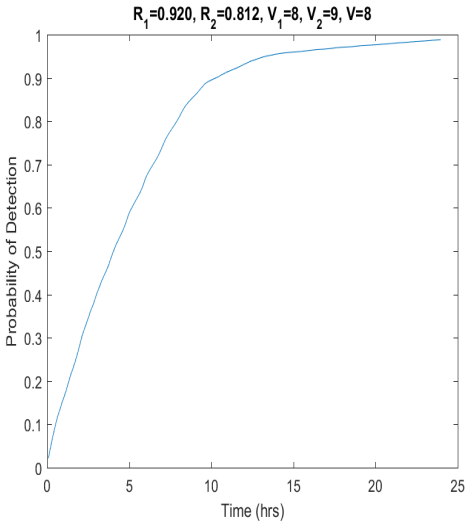


Figure A.469. P_D vs Time

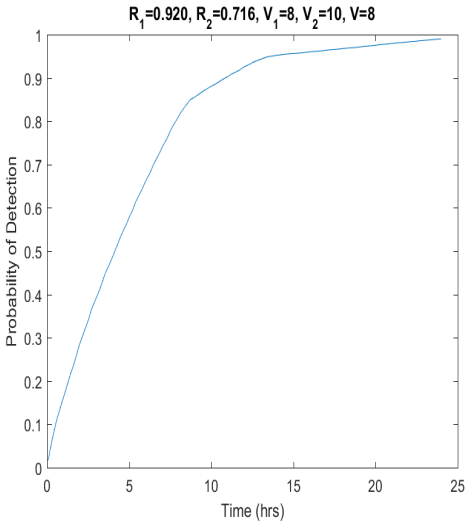


Figure A.470. P_D vs Time

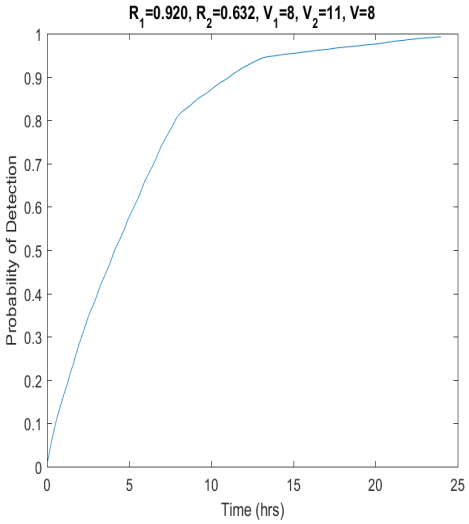


Figure A.471. P_D vs Time

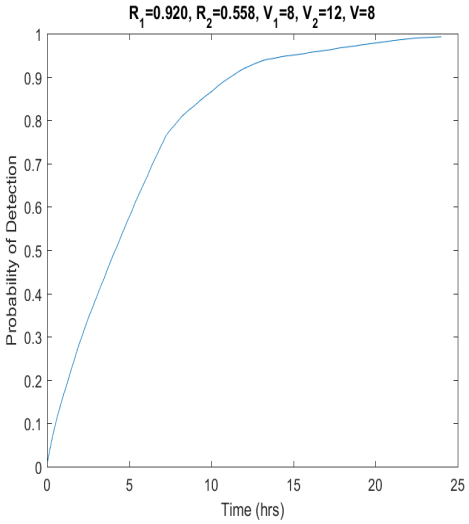


Figure A.472. P_D vs Time

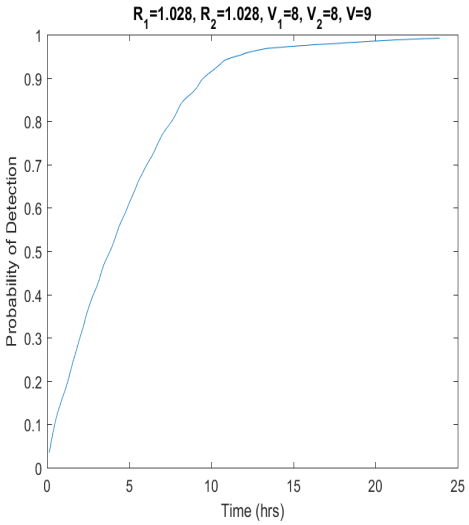


Figure A.473. P_D vs Time

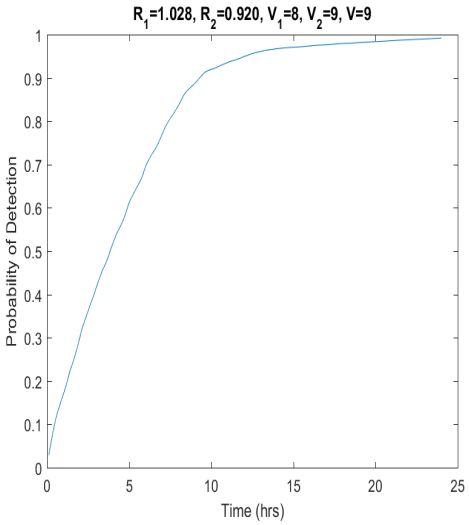


Figure A.474. P_D vs Time

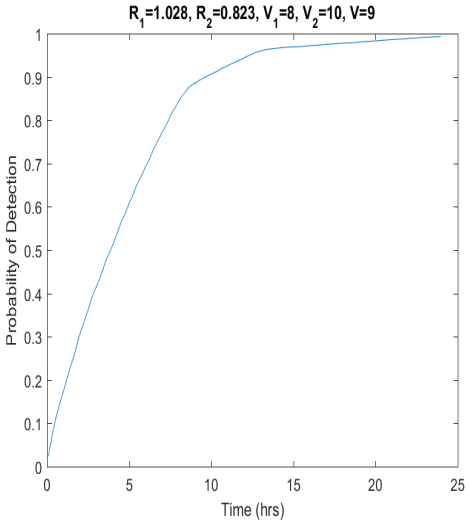


Figure A.475. P_D vs Time

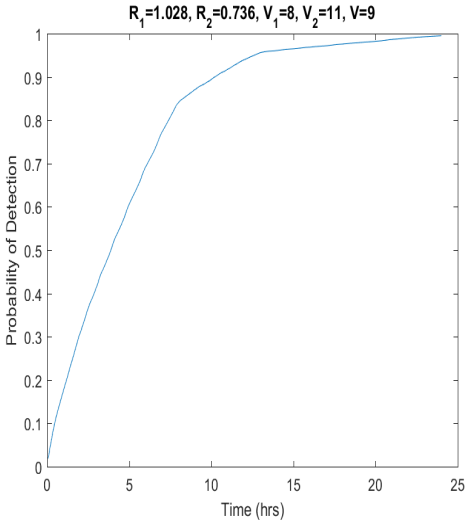


Figure A.476. P_D vs Time

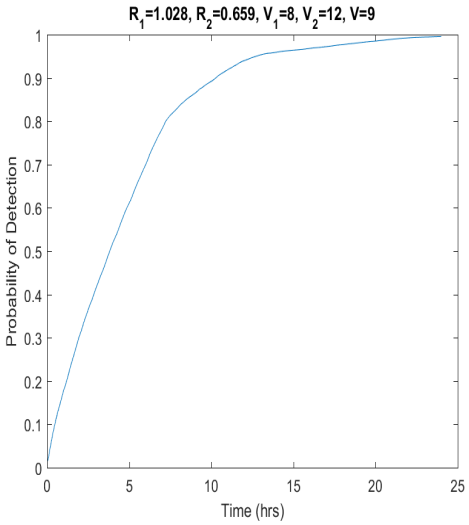


Figure A.477. P_D vs Time

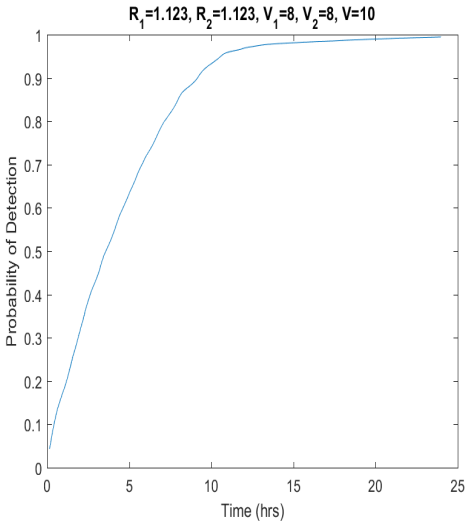


Figure A.478. P_D vs Time

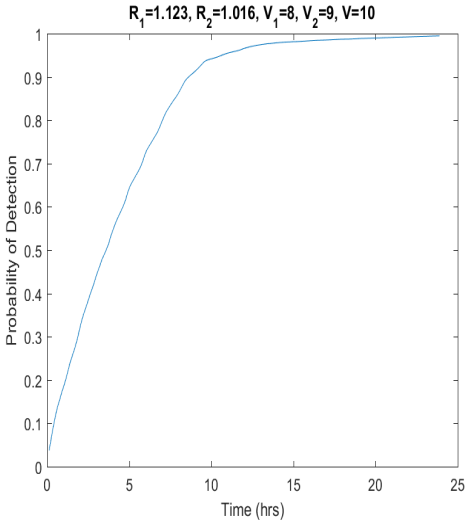


Figure A.479. P_D vs Time

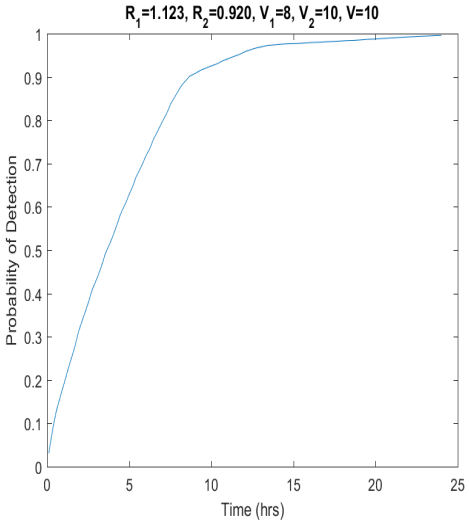


Figure A.480. P_D vs Time

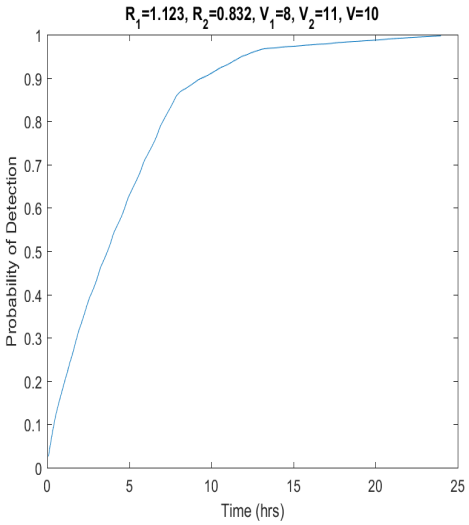


Figure A.481. P_D vs Time

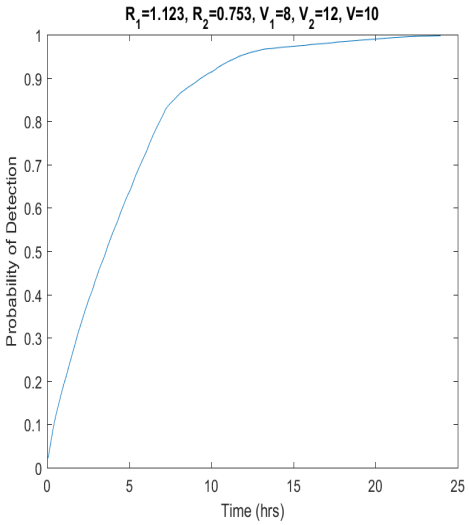


Figure A.482. P_D vs Time

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B:
Tables

B.1 Discrete

Table B.1. Parameter Variation for p_i for Uniform Distribution

Parameter	b	r^2	SSE	RMSE
$p_{uniform}$,$q_{uniform}$	0.01061	0.99242	0.006608	0.027096
$p_{uniform}$,q_{long}	0.009988	0.99935	0.00049844	0.007442
$p_{uniform}$,q_{corner}	0.01198	0.99795	0.0015533	0.013137

Best fit line parameters for a uniformly distributed location probability density of the form $f(x) = 1 - e^{-bx}$

Table B.2. Parameter Variation for p_i for Corner Stratification

Parameter	b	r^2	SSE	RMSE
$p_{cornerstrat}$,$q_{uniform}$	0.01321	0.99908	0.00065853	0.008554
$p_{cornerstrat}$,q_{long}	0.00896	0.9992	0.00066179	0.0085751
$p_{cornerstrat}$,q_{corner}	0.01119	0.99808	0.0013515	0.012254

Best fit line parameters for a corner stratified distribution location probability density of the form $f(x) = 1 - e^{-bx}$

Table B.3. Parameter Variation for p_i for Centered Gaussian

Parameter	b	r^2	SSE	RMSE
p_{Gauss},$q_{uniform}$	0.0147	0.99781	0.0014554	0.012717
p_{Gauss},q_{long}	0.01315	0.99609	0.0026167	0.017051
p_{Gauss},q_{corner}	0.01589	0.9963	0.00234	0.016124

Best fit line parameters for a centered Gaussian distribution location probability density of the form $f(x) = 1 - e^{-bx}$

Table B.4. Parameter Variation for p_i for Side Gaussian

Parameter	b	r^2	SSE	RMSE
p_{Gauss},$q_{uniform}$	0.02405	0.9944	0.0027756	0.017561
p_{Gauss},q_{long}	0.01428	0.99225	0.0047039	0.0228621
p_{Gauss},q_{corner}	0.01895	0.98981	0.0054218	0.024544

Best fit line parameters for a offset Gaussian distribution location probability density of the form $f(x) = 1 - e^{-bx}$

B.2 Bayesian

Table B.5. Parameter Variation for p_i for Uniform Distribution

Parameter	a	b	r^2	SSE	RMSE
p_{strat},$q_{uniform}$	0.90643	-0.01139	0.99645	0.004106	0.015103
p_{strat},q_{long}	0.89218	-0.011534	0.98964	0.0104	0.024037
p_{strat},q_{corner}	0.8876	-0.014151	0.99384	0.0058881	0.018086

Best fit line parameters for a uniformly distributed location probability density of the form $f(x) = a - e^{-bx}$

Table B.6. Parameter Variation for p_i for Corner Stratification

Parameter	a	b	r^2	SSE	RMSE
p_{strat},$q_{uniform}$	0.88918	-0.018268	0.976	0.0175	0.031181
p_{strat},q_{long}	0.89054	-0.012323	0.98204	0.016665	0.030427
p_{strat},q_{corner}	0.89926	-0.011051	0.99147	0.008804	0.022116

Best fit line parameters for a corner stratified distribution location probability density of the form $f(x) = a - e^{-bx}$

Table B.7. Parameter Variation for p_i for Centered Gaussian

Parameter	a	b	r^2	SSE	RMSE
p_{strat},$q_{uniform}$	0.88971	-0.017384	0.9847	0.012037	0.025859
p_{strat},q_{long}	0.88274	-0.016095	0.98036	0.01587	0.029693
p_{strat},q_{corner}	0.88519	-0.019317	0.98268	0.01265	0.02651

Best fit line parameters for a centered Gaussian distribution location probability density of the form $f(x) = a - e^{-bx}$

Table B.8. Parameter Variation for p_i for Side Gaussian

Parameter	a	b	r^2	SSE	RMSE
p_{strat},$q_{uniform}$	0.88957	-0.029116	0.97623	0.012426	0.026275
p_{strat},q_{long}	0.88243	-0.01768	0.96822	0.023033	0.035772
p_{strat},q_{corner}	0.88301	-0.023588	0.96697	0.019693	0.0330771

Best fit line parameters for a offset Gaussian distribution location probability density of the form $f(x) = a - e^{-bx}$

B.3 Continuous

B.3.1 Inline Ladder Search

Table B.9. Inline Ladder Search: Time to $P_D = 90\%(V = 5)$

		Ship 1				
	Velocities	8	9	10	11	12
	8	14.8081	14.5798	15.6685	16.1428	16.6395
	9	15.7111	15.6258	16.6558	16.8813	17.6232
Ship 2	10	18.5017	18.0867	17.8148	18.2146	18.7484
	11	18.2434	19.5095	19.5958	18.8285	20.0086
	12	18.1569	19.1277	20.5486	21.9053	21.3588

Inline Ladder Search

Table B.10. Inline Ladder Search: Time to $P_D = 90\%(V = 6)$

		Ship 1				
	Velocities	8	9	10	11	12
	8	12.7567	12.2523	12.1515	12.4612	13.1644
	9	13.3414	12.6024	12.3187	12.5611	13.1930
Ship 2	10	14.0466	13.0154	12.6810	13.2800	13.5936
	11	13.7793	15.2770	14.1587	14.0189	14.6239
	12	13.9657	15.0961	16.4268	15.6398	15.5540

Inline Ladder Search

Table B.11. Inline Ladder Search: Time to $P_D = 90\%(V = 7)$

		Ship 1				
	Velocities	8	9	10	11	12
	8	10.9647	10.6280	10.4097	10.2503	10.2367
	9	11.7028	11.2848	10.7237	10.5538	10.5661
Ship 2	10	12.7877	11.5763	11.0378	10.8235	10.7182
	11	12.2397	12.4420	11.3968	11.0259	10.9716
	12	11.8331	12.0358	12.8213	11.4023	11.4783

Inline Ladder Search

Table B.12. Inline Ladder Search: Time to $P_D = 90\%(V = 8)$

		Ship 1				
	Velocities	8	9	10	11	12
	8	8.7756	8.6460	8.8498	8.7901	8.8294
	9	10.1282	9.7988	9.2987	9.2396	9.0239
Ship 2	10	11.4149	10.2606	9.8759	9.5892	9.3740
	11	11.1874	11.3872	10.2884	9.8888	9.6073
	12	10.9687	11.0076	11.0854	10.1519	9.7240

Inline Ladder Search

Table B.13. Inline Ladder Search: Time to $P_D = 90\%(V = 9)$

		Ship 1				
	Velocities	8	9	10	11	12
	8	6.8662	6.9782	7.1117	7.5250	7.7083
	9	7.5069	7.8241	7.7043	7.8639	7.9255
Ship 2	10	9.7362	9.0589	8.7202	8.4741	8.3054
	11	10.1011	10.0333	9.2198	8.9486	8.6854
	12	10.0425	10.1511	10.0968	9.2826	8.9569

Inline Ladder Search

Table B.14. Inline Ladder Search: Time to $P_D = 90\%(V = 10)$

		Ship 1				
	Velocities	8	9	10	11	12
	8	6.6247	6.4570	6.3438	6.4058	6.5204
	9	6.5861	6.3278	6.3438	6.4924	6.8038
Ship 2	10	7.2954	6.9782	7.0840	7.0118	7.1582
	11	9.1759	8.6565	8.2237	7.8774	7.6543
	12	9.2135	9.2844	9.0009	8.3631	8.1504

Inline Ladder Search

B.3.2 Multipath Ladder

B.3.3 Multipath Ladder Search

Table B.15. Multipath Ladder Search: Time to $P_D = 90\%(V = 5)$

		Ship 1				
	Velocities	8	9	10	11	12
	8	14.6171	14.9854	15.6398	16.1907	16.5752
	9	14.9427	15.8392	16.3410	16.9964	17.6940
Ship 2	10	18.2155	17.4714	17.5716	18.2338	18.7098
	11	18.2913	19.0874	18.5983	18.9148	19.9443
	12	18.3240	19.1792	20.6129	20.6580	21.1080

Multipath Ladder Search

Table B.16. Multipath Ladder Search: Time to $P_D = 90\%(V = 6)$

		Ship 1				
	Velocities	8	9	10	11	12
	8	12.7024	12.2912	12.2072	13.0803	13.1644
	9	12.9135	12.4857	12.4580	12.9206	13.2073
Ship 2	10	13.7122	12.7089	12.7089	13.2600	13.7510
	11	14.0389	14.0988	13.6395	14.2585	14.5953
	12	13.9371	15.6255	15.8545	15.2106	15.5254

Multipath Ladder Search

Table B.17. Multipath Ladder Search: Time to $P_D = 90\%(V = 7)$

		Ship 1				
	Velocities	8	9	10	11	12
	8	11.4414	10.9266	10.4994	10.2503	10.3888
	9	11.8222	11.1654	10.7686	10.6212	10.7942
Ship 2	10	12.3390	11.3070	10.9481	10.8235	11.0729
	11	12.4083	11.7002	11.1270	10.8910	11.3263
	12	12.0611	12.3145	11.5037	11.0729	11.630

Multipath Ladder Search

Table B.18. Multipath Ladder Search: Time to $P_D = 90\%(V = 8)$

		Ship 1				
	Velocities	8	9	10	11	12
	8	10.1501	9.7165	9.2987	9.0398	8.9850
	9	10.7870	10.2105	9.7476	9.2895	9.1406
Ship 2	10	11.2226	10.5813	9.9400	9.5392	9.45180
	11	11.5370	10.7379	10.0886	9.7890	9.6073
	12	11.0854	11.2021	10.3464	9.8407	9.6462

Multipath Ladder Search

Table B.19. Multipath Ladder Search: Time to $P_D = 90\%(V = 9)$

		Ship 1				
	Velocities	8	9	10	11	12
	8	8.1866	8.2470	8.2123	8.1351	8.0883
	9	9.3043	8.9871	8.7202	8.3385	8.1969
Ship 2	10	10.3288	9.5669	9.2282	8.7452	8.4140
	11	10.7790	10.0333	9.4231	9.0164	8.6854
	12	10.4768	10.5311	9.7168	9.1197	8.7940

Multipath Ladder Search

Table B.20. Multipath Ladder Search: Time to $P_D = 90\%(V = 10)$

		Ship 1				
	Velocities	8	9	10	11	12
	8	6.7825	6.8444	6.7668	7.0983	7.0873
	9	6.9735	7.4901	7.4011	7.5312	7.3708
Ship 2	10	8.7756	8.4585	8.2470	7.8774	7.6543
	11	9.9550	9.1759	8.6565	8.3102	8.0087
	12	9.8514	9.8514	9.0009	8.5757	8.221

Multipath Ladder Search

B.3.4 Spiral

B.3.5 Spiral Search

Table B.21. Spiral Search: Time to $P_D = 90\%(V = 5)$

		Ship 1				
	Velocities	8	9	10	11	12
	8	19.0117	19.4895	19.5176	19.7109	20.2015
	9	21.6669	20.4074	20.8340	20.8907	20.0729
Ship 2	10	22.2220	23.1092	21.5638	21.5909	21.67388
	11	21.8115	22.8090	25.0000	22.5980	22.65114
	12	21.7960	22.4003	23.4676	25.0000	25.0000

Spiral Search

Table B.22. Spiral Search: Time to $P_D = 90\%(V = 6)$

		Ship 1				
	Velocities	8	9	10	11	12
	8	13.8424	14.3527	14.3253	14.5182	16.0405
	9	16.0252	15.7918	16.2205	16.5751	15.0818
Ship 2	8	18.3108	17.6140	17.1124	17.3739	17.5859
	11	18.2326	18.7518	18.6519	18.2126	18.1725
	12	18.3013	18.9309	19.5462	19.8467	18.9309

Spiral Search

Table B.23. Spiral Search: Time to $P_D = 90\%(V = 7)$

		Ship 1				
	Velocities	8	9	10	11	12
	8	11.5209	11.8819	12.1147	12.1723	12.5932
	9	11.7625	12.0013	12.3839	12.1723	11.4023
Ship 2	10	15.1658	13.7749	13.5505	13.7233	14.0122
	11	15.0720	15.8476	15.0383	14.8697	15.1017
	12	14.8737	16.1153	16.5460	15.9886	15.7099

Spiral Search

Table B.24. Spiral Search: Time to $P_D = 90\%(V = 8)$

		Ship 1				
	Velocities	8	9	10	11	12
	8	10.3616	10.2929	10.9019	11.0875	11.2021
	9	10.4576	10.2929	10.6454	10.6879	10.5409
Ship 2	10	11.8638	10.5813	10.6454	10.7878	10.6575
	11	12.2362	13.2850	11.9365	11.8866	11.7855
	12	12.0189	13.6137	13.9637	13.1469	12.9524

Spiral Search

Table B.25. Spiral Search: Time to $P_D = 90\%(V = 9)$

		Ship 1				
	Velocities	8	9	10	11	12
Ship 2	8	9.6390	9.3043	9.6515	10.2367	10.3139
	9	9.8330	9.4100	9.3129	9.7621	9.8254
	10	10.4135	9.4822	9.3129	9.5587	9.6625
	11	10.5078	10.0333	9.5587	9.6943	9.71684
	12	10.3682	11.6710	11.5625	10.6397	10.5854

Spiral Search

Table B.26. Spiral Search: Time to $P_D = 90\%(V = 10)$

		Ship 1				
	Velocities	8	9	10	11	12
Ship 2	8	9.3062	8.6524	8.6699	9.5221	9.4262
	9	9.2981	8.7815	8.5642	8.8296	9.2135
	10	9.6215	8.8814	8.5642	8.5699	8.8592
	11	9.6087	9.1759	8.5699	8.6565	8.7883
	12	9.3553	9.7805	9.0009	8.7174	8.788

Spiral Search

APPENDIX C:
Codes

C.1 Discrete

C.1.1 Main Code

```
clc
clear
discrete_search_prob
discrete_loc_probs2
close all
%p-uniform
dis_pUni_qUni
dis_pUni_qlongstrat
dis_pUni_qbothstrat
%p-long stratified
dis_plongstrat_qUni
dis_plongstrat_qlongstrat
dis_plongstrat_qbothstrat
%p-both stratified
dis_pbothstrat_qUni
dis_pbothstrat_qlongstrat
dis_pbothstrat_qbothstrat
%p-normal center
dis_pnormwide_qUni
dis_normwide_qlongstrat
dis_normwide_qbothstrat
%p-normal side
dis_pnormside_qUni
dis_normside_qlongstrat
dis_normside_qbothstrat
```

```
%p-normal corner
dis_pnormc_qUni
dis_normc_qlongstrat
dis_normc_qbothstrat
%%

pd=[Pd_uni_uni(end),Pd_uni_ls(end),Pd_uni_bs(end),Pd_ls_uni(end),...
    Pd_ls_ls(end),Pd_ls_bs(end),Pd_bs_uni(end),Pd_bs_ls(end),Pd_bs_bs(end)...
    ,Pd_nw_uni(end),Pd_nw_ls(end),Pd_nw_bs(end),Pd_ns_uni(end),...
    Pd_ns_ls(end),Pd_ns_bs(end),Pd_nc_uni(end),Pd_nc_ls(end),...
    Pd_nc_bs(end)]

%%
% close all
set(0,'defaulttextinterpreter','latex')
clear g
[ ~,~,~,Pd_uni_uni,k] = discrete_search2(P_uni,q_uni,500, 1);
[ ~,~,~,Pd_uni_ls ,k] = discrete_search2(P_uni,q_long_strat,500, 1 );
[ ~,~,~,Pd_uni_bs ,k] = discrete_search2(P_uni,q_both_strat,500, 1 );
figure

step1=10:round(.05*length(Pd_uni_uni)):length(Pd_uni_uni);
step2=10:round(.05*length(Pd_uni_ls)):length(Pd_uni_ls);
step3=10:round(.05*length(Pd_uni_bs)):length(Pd_uni_bs);
plot(step1,Pd_uni_uni(step1),'xk',step2,Pd_uni_ls(step2),'xr',step3,Pd_uni_bs(step3),'xg')
```

```
51 g = fittype('1-exp(b*x)','dependent','Prob','independent','x
   ');
52
53 [fit_bb1,gof_bb1]=fit(step1',Pd_uni_uni(step1)',g,'
   StartPoint',[0],'Upper',[0]);
54 [fit_bb2,gof_bb2]=fit(step2',Pd_uni_ls(step2)',g,'StartPoint
   ',[0],'Upper',[0]);
55 [fit_bb3,gof_bb3]=fit(step3',Pd_uni_bs(step3)',g,'StartPoint
   ',[0],'Upper',[0]);
56 hold on
57 plot(fit_bb1,'k')
58 plot(fit_bb2,'r')
59 plot(fit_bb3,'g')
60 xlabel('n')
61 ylabel('$P_{D}$')
62 title('$P_{D}$ vs Number of Glimpses and Exponential Fitted
   Curves')
63 legend('(p_{uniform},q_{uniform})','(p_{uniform},q_{strat a
   })','(p_{uniform},q_{strat b})'...
64     ,sprintf('1-e^{%.5f*x}',fit_bb1.b),sprintf('1-e^{%.5f*x}
       ',fit_bb2.b),...
65     sprintf('1-e^{%.5f*x}',fit_bb3.b),'location','best')
66 print('Bayes_Uni_Fit','-dpng')
67
68 [~,~,~,Pd_ls_uni,k] = discrete_search2(P_long_strat,q_uni
   ,500, 1 );
69 [~,~,~,Pd_ls_ls ,k] = discrete_search2(P_long_strat,
   q_long_strat,500, 1 );
70 [~,~,~,Pd_ls_bs ,k] = discrete_search2(P_long_strat,
   q_both_strat,500, 1 );
71 figure
72 step1=10:round(.05*length(Pd_ls_uni)):length(Pd_ls_uni);
73 step2=10:round(.05*length(Pd_ls_ls)):length(Pd_ls_ls);
```

```
step3=10:round(.05*length(Pd_ls_bs)):length(Pd_ls_bs);
plot(step1,Pd_ls_uni(step1),'xk',step2,Pd_ls_ls(step2),'xr',
    step3,Pd_ls_bs(step3),'xg')
[fit_bb4,gof_bb4]=fit(step1',Pd_ls_uni(step1)',g,'StartPoint
    ',[0],'Upper',[0]);
[fit_bb5,gof_bb5]=fit(step2',Pd_ls_ls(step2)',g,'StartPoint'
    ,[0],'Upper',[0]);
[fit_bb6,gof_bb6]=fit(step3',Pd_ls_bs(step3)',g,'StartPoint'
    ,[0],'Upper',[0]);
hold on
plot(fit_bb4,'k')
plot(fit_bb5,'r')
plot(fit_bb6,'g')
xlabel('n')
ylabel('$P_{D}$')
title('$P_{D}$ vs Number of Glimpses and Exponential Fitted
    Curves')
legend('(p_{strat a},q_{uniform})','(p_{strat a},q_{strat a
    })','(p_{strat a},q_{strat b})'...
     ,sprintf('1-e^{%.5f*x}',fit_bb4.b),sprintf('1-e^{%.5f*x}
        ',fit_bb5.b),...
     sprintf('1-e^{%.5f*x}',fit_bb6.b),'location','best')
print('Bayes_Long_Fit','-dpng')

[~,~,~,Pd_bs_uni,k] = discrete_search2(P_both_strat,q_uni
    ,500, 1 );
[~,~,~,Pd_bs_ls ,k] = discrete_search2(P_both_strat,
    q_long_strat,500, 1 );
[~,~,~,Pd_bs_bs ,k] = discrete_search2(P_both_strat,
    q_both_strat,500, 1 );
figure
step1=10:round(.05*length(Pd_bs_uni)):length(Pd_bs_uni);
step2=10:round(.05*length(Pd_bs_ls)):length(Pd_bs_ls);
```

```
97  step3=10:round(.05*length(Pd_bs_bs)):length(Pd_bs_bs);
98  plot(step1,Pd_bs_uni(step1),'xk',step2,Pd_bs_ls(step2),'xr',
        step3,Pd_bs_bs(step3),'xg')
99  [fit_bb7,gof_bb7]=fit(step1',Pd_bs_uni(step1)',g,'StartPoint
        ',[0],'Upper',[0]);
100 [fit_bb8,gof_bb8]=fit(step2',Pd_bs_ls(step2)',g,'StartPoint'
        ,[0],'Upper',[0]);
101 [fit_bb9,gof_bb9]=fit(step3',Pd_bs_bs(step3)',g,'StartPoint'
        ,[0],'Upper',[0]);
102 hold on
103 plot(fit_bb7,'k')
104 plot(fit_bb8,'r')
105 plot(fit_bb9,'g')
106 xlabel('n')
107 ylabel('$P_{D}$')
108 title('$P_{D}$ vs Number of Glimpses and Exponential Fitted
        Curves')
109 legend('(p_{strat b},q_{uniform})','(p_{strat b},q_{strat a
        })','(p_{strat b},q_{strat b})'...
110       ,sprintf('1-e^{%.5f*x}',fit_bb7.b),sprintf('1-e^{%.5f*x}
            ',fit_bb8.b),...
111       sprintf('1-e^{%.5f*x}',fit_bb9.b),'location','best')
112 print('Bayes_Both_Fit','-dpng')
113
114 [~,~,~,Pd_nw_uni,k] = discrete_search2(P_norm_wide,q_uni
        ,500, 1 );
115 [~,~,~,Pd_nw_ls ,k] = discrete_search2(P_norm_wide,
        q_long_strat,500, 1 );
116 [~,~,~,Pd_nw_bs ,k] = discrete_search2(P_norm_wide,
        q_both_strat,500, 1 );
117 figure
118 step1=10:round(.05*length(Pd_nw_uni)):length(Pd_nw_uni);
119 step2=10:round(.05*length(Pd_nw_ls)):length(Pd_nw_ls);
```

```
step3=10:round(.05*length(Pd_nw_bs)):length(Pd_nw_bs);
plot(step1,Pd_nw_uni(step1),'xk',step2,Pd_nw_ls(step2),'xr',
    step3,Pd_nw_bs(step3),'xg')
[fit_bb10,gof_bb10]=fit(step1',Pd_nw_uni(step1)',g,'
    StartPoint',[0],'Upper',[0]);
[fit_bb11,gof_bb11]=fit(step2',Pd_nw_ls(step2)',g,'
    StartPoint',[0],'Upper',[0]);
[fit_bb12,gof_bb12]=fit(step3',Pd_nw_bs(step3)',g,'
    StartPoint',[0],'Upper',[0]);
hold on
plot(fit_bb10,'k')
plot(fit_bb11,'r')
plot(fit_bb12,'g')
xlabel('n')
ylabel('$P_{D}$')
title('$P_{D}$ vs Number of Glimpses and Exponential Fitted
    Curves')
legend('(p_{norm},q_{uniform})','(p_{norm},q_{strat a})','(
    p_{norm},q_{strat b})'...
    ,sprintf('1-e^{%.5f*x}',fit_bb10.b),sprintf('1-e^{%.5f*x
        }',fit_bb11.b),...
    sprintf('1-e^{%.5f*x}',fit_bb12.b),'location','best')
print('Bayes_Norm_Fit','-dpng')

[~,~,~,Pd_ns_uni,k] = discrete_search2(P_norm_side,q_uni
    ,500, 1 );
[~,~,~,Pd_ns_ls ,k] = discrete_search2(P_norm_side,
    q_long_strat,500, 1 );
[~,~,~,Pd_ns_bs ,k] = discrete_search2(P_norm_side,
    q_both_strat,500, 1 );
figure
step1=10:round(.05*length(Pd_ns_uni)):length(Pd_ns_uni);
step2=10:round(.05*length(Pd_ns_ls)):length(Pd_ns_ls);
```

```
step3=10:round(.05*length(Pd_ns_bs)):length(Pd_ns_bs);
plot(step1,Pd_ns_uni(step1),'xk',step2,Pd_ns_ls(step2),'xr',
    step3,Pd_ns_bs(step3),'xg')
[fit_bb13,gof_bb13]=fit(step1',Pd_ns_uni(step1)',g,'
    StartPoint',[0],'Upper',[0]);
[fit_bb14,gof_bb14]=fit(step2',Pd_ns_ls(step2)',g,'
    StartPoint',[0],'Upper',[0]);
[fit_bb15,gof_bb15]=fit(step3',Pd_ns_bs(step3)',g,'
    StartPoint',[0],'Upper',[0]);
hold on
plot(fit_bb13,'k')
plot(fit_bb14,'r')
plot(fit_bb15,'g')
xlabel('n')
ylabel('$P_{D}$')
title('$P_{D}$ vs Number of Glimpses and Exponential Fitted
    Curves')
legend('(p_{norm b},q_{uniform})','(p_{norm b},q_{strat a})'
    ,'(p_{norm b},q_{strat b})'...
     ,sprintf('1-e^{%.5f*x}',fit_bb13.b),sprintf('1-e^{%.5f*x
        }',fit_bb14.b),...
     sprintf('1-e^{%.5f*x}',fit_bb15.b),'location','best')
print('Bayes_NormSide_Fit','-dpng')

[~,~,~,Pd_nc_uni,k] = discrete_search2(P_norm_corner,q_uni
    ,500, 1 );
[~,~,~,Pd_nc_ls ,k] = discrete_search2(P_norm_corner,
    q_long_strat,500, 1 );
[~,~,~,Pd_nc_bs ,k] = discrete_search2(P_norm_corner,
    q_both_strat,500, 1 );
figure
step1=10:round(.05*length(Pd_nc_uni)):length(Pd_nc_uni);
step2=10:round(.05*length(Pd_nc_ls)):length(Pd_nc_ls);
```

```
166 step3=10:round(.05*length(Pd_nc_bs)):length(Pd_nc_bs)
167 plot(step1,Pd_nc_uni(step1),'xk',step2,Pd_nc_ls(step2),'xr',
      step3,Pd_nc_bs(step3),'xg')
168 [fit_bb16,gof_bb16]=fit(step1',Pd_nc_uni(step1)',g,'
      StartPoint',[0],'Upper',[0]);
169 [fit_bb17,gof_bb17]=fit(step2',Pd_nc_ls(step2)',g,'
      StartPoint',[0],'Upper',[0]);
170 [fit_bb18,gof_bb18]=fit(step3',Pd_nc_bs(step3)',g,'
      StartPoint',[0],'Upper',[0]);
171 hold on
172 plot(fit_bb16,'k')
173 plot(fit_bb17,'r')
174 plot(fit_bb18,'g')
175 xlabel('n')
176 ylabel('$P_{D}$')
177 title('$P_{D}$ vs Number of Glimpses and Exponential Fitted
      Curves')
178 legend('(p_{norm c},q_{uniform})','(p_{norm c},q_{strat a})'
      ,'(p_{norm c},q_{strat b})'...
179       ,sprintf('1-e^{%.5f*x}',fit_bb16.b),sprintf('1-e^{%.5f*x
            }',fit_bb17.b),...
180       sprintf('1-e^{%.5f*x}',fit_bb18.b),'location','best')
181 print('Bayes_NormCorner_Fit','-dpng')
182 format shortG
183 b=[fit_bb1.b,fit_bb2.b,fit_bb3.b,fit_bb4.b,fit_bb5.b,fit_bb6
      .b,fit_bb7.b,...
184       fit_bb8.b,fit_bb9.b,fit_bb10.b,fit_bb11.b,fit_bb12.b,
            fit_bb13.b,fit_bb14.b,...
185       fit_bb15.b,fit_bb16.b,fit_bb17.b,fit_bb18.b]
186 r2=[gof_bb1.rsquare,gof_bb2.rsquare,gof_bb3.rsquare,gof_bb4.
      rsquare,...
187       gof_bb5.rsquare,gof_bb6.rsquare,gof_bb7.rsquare,gof_bb8.
            rsquare,...
```

```
188       gof_bb9.rsquare,gof_bb10.rsquare,gof_bb11.rsquare,
              gof_bb12.rsquare,...
189       gof_bb13.rsquare,gof_bb14.rsquare,gof_bb15.rsquare,
              gof_bb16.rsquare,...
190       gof_bb17.rsquare,gof_bb18.rsquare]
```

C.1.2 Search Probabilities q_i

```
1  format short
2  format compact
3  Area=1000;
4  x=-5:5;y=x;
5  [X,Y]=meshgrid(x,y);
6  %%
7  q0=.3;
8  q_uni=q0*ones(10,10);
9
10 figure
11 subplot(1,3,1)
12 imagesc([-4.5,4.5],[-4.5,4.5],q_uni,[0,1])
13 set(gca, 'YDir', 'normal')
14 cb_q=colorbar;
15 title(cb_q,'q_{i}')
16 hold on
17 plot(X,Y,'k');
18 plot(Y,X,'k');
19 title('Uniform')
20 xlabel('Long')
21 ylabel('Lat')
22 %%
23 %
24 q0=.07;
25 qx=1:10;
26 qy=ones(10,1);
```

```
q_long_strat=qy*qx*q0*.9;
subplot(1,3,2)
imagesc([-4.5,4.5],[-4.5,4.5],q_long_strat,[0,max(
    q_long_strat(:))])
set(gca, 'YDir', 'normal')
cb_q=colorbar;
title(cb_q,'q_{i}')
hold on
plot(X,Y,'k');
plot(Y,X,'k');
title('Strat. a')
xlabel('Long')
ylabel('Lat')
%%
q0=.01;
qx=(1:10);
qy=(1:10)';
q_both_strat=qy*qx*q0*.9;
subplot(1,3,3)
imagesc([-4.5,4.5],[-4.5,4.5],q_both_strat,[0,max(
    q_both_strat(:))])
set(gca, 'YDir', 'normal')
cb_q=colorbar;
title(cb_q,'q_{i}')
hold on
plot(X,Y,'k');
plot(Y,X,'k');
title('Strat. b')
xlabel('Long')
ylabel('Lat')
%%
print('Discrete_Miss_Probablities','-dpng')
```

C.1.3 Location Probabilities p_i

```
% clear
close all
clc
format short
format compact
Area=1000;
%create the grid positions
p0=0;
x=-5:5;y=x;
[X,Y]=meshgrid(x,y);
%%
P_uni=ones(10,10);
P_uni=(1-p0)*P_uni/sum(P_uni(:));
figure
subplot(2,3,1)
imagesc([-4.5,4.5],[-4.5,4.5],P_uni,[0,1])
set(gca, 'YDir', 'normal')
cb_p=colorbar;
title(cb_p,'p_{i}')
hold on
plot(X,Y,'k');
plot(Y,X,'k');
title('Uniform')
xlabel('Long.')
ylabel('Lat')
%%
Px=1:10;
Py=ones(10,1);
P_long_strat=Py*Px;
P_long_strat=(1-p0)*P_long_strat/sum(P_long_strat(:));
subplot(2,3,2)
imagesc([-4.5,4.5],[-4.5,4.5],P_long_strat,[0,max(
```

```
    P_long_strat(:))])
set(gca, 'YDir', 'normal')
cb_p=colorbar;
title(cb_p,'p_{i}')
hold on
plot(X,Y,'k');
plot(Y,X,'k');
title('Stratified a')
xlabel('Long.')
ylabel('Lat')
%%
Px=1:10;
Py=(1:10)';
P_both_strat=Py*Px;
P_both_strat=(1-p0)*P_both_strat/sum(P_both_strat(:));
% figure
subplot(2,3,3)
imagesc([-4.5,4.5],[-4.5,4.5],P_both_strat,[0,max(
    P_both_strat(:))])
set(gca, 'YDir', 'normal')
cb_p=colorbar;
title(cb_p,'p_{i}')
hold on
plot(X,Y,'k');
plot(Y,X,'k');
title('Stratified b')
xlabel('Long.')
ylabel('Lat')
%%
Px = normpdf([-4.5:1:4.5],0,2);
Py = normpdf([-4.5:1:4.5],0,5)';
P_norm_wide=Py*Px;
P_norm_wide=(1-p0)*P_norm_wide/sum(P_norm_wide(:));
```

```
64 subplot(2,3,4)
65 imagesc([-4.5,4.5],[-4.5,4.5],P_norm_wide,[0,max(P_norm_wide
       (:))])
66 set(gca, 'YDir', 'normal')
67 cb_p=colorbar;
68 title(cb_p,'p_{i}')
69 hold on
70 plot(X,Y,'k');
71 plot(Y,X,'k');
72 title('Normal a')
73 xlabel('Long.')
74 ylabel('Lat')
75 %%
76 Px = normpdf([-4.5:1:4.5],3,2);
77 Py = normpdf([-4.5:1:4.5],3,2)';
78 P_norm_corner=Py*Px;
79 P_norm_corner=(1-p0)*P_norm_corner/sum(P_norm_corner(:));
80 subplot(2,3,5)
81 imagesc([-4.5,4.5],[-4.5,4.5],P_norm_corner,[0,max(
       P_norm_corner(:))])
82 set(gca, 'YDir', 'normal')
83 cb_p=colorbar;
84 title(cb_p,'p_{i}')
85 hold on
86 plot(X,Y,'k');
87 plot(Y,X,'k');
88 title('Normal b')
89 xlabel('Long.')
90 ylabel('Lat')
91 %%
92 Px = normpdf([-4.5:1:4.5],3,2);
93 Py = normpdf([-4.5:1:4.5],0,2)';
94 P_norm_side=Py*Px;
```

```
P_norm_side=(1-p0)*P_norm_side/sum(P_norm_side(:));
subplot(2,3,6)
imagesc([-4.5,4.5],[-4.5,4.5],P_norm_side,[0,max(P_norm_side
    (:))])
set(gca, 'YDir', 'normal')
cb_p=colorbar;
title(cb_p,'p_{i}')
hold on
plot(X,Y,'k');
plot(Y,X,'k');
title('Normal c')
xlabel('Long.')
ylabel('Lat')
%%
print('Discrete_Location_Probablities','-dpng')
```

C.1.4 Sample code for the Uniform/Uniform case

```
[ pc,n,imax,Pd_uni_uni ,k] = discrete_search2(P_uni,q_uni
    ,50, .95 );
figure
subplot(2,2,1)
imagesc([-4.5,4.5],[-4.5,4.5],pc,[0,max(pc(:))])
set(gca, 'YDir', 'normal')
cb_p=colorbar;
title(cb_p,'p_{i}')
hold on
plot(X,Y,'k');
plot(Y,X,'k');
title('Remainder Prob.')
xlabel('Long.')
ylabel('Lat.')

subplot(2,2,2)
```

```
16 imagesc([-4.5,4.5],[-4.5,4.5],n,[0,max(n(:))])
17 set(gca, 'YDir', 'normal')
18 cb_p=colorbar;
19 title(cb_p,'N')
20 hold on
21 plot(X,Y,'k');
22 plot(Y,X,'k');
23 title('Actual Search')
24 xlabel('Long.')
25 ylabel('Lat.')
26 subplot(2,2,3)
27 imagesc([-4.5,4.5],[-4.5,4.5],P_uni,[0,max(P_uni(:))])
28 set(gca, 'YDir', 'normal')
29 cb_p=colorbar;
30 title(cb_p,'p_{i}')
31 hold on
32 plot(X,Y,'k');
33 plot(Y,X,'k');
34 title('Uniform')
35 xlabel('Long.')
36 ylabel('Lat.')
37
38
39 subplot(2,2,4)
40 imagesc([-4.5,4.5],[-4.5,4.5],q_uni,[0,max(q_uni(:))])
41 set(gca, 'YDir', 'normal')
42 cb_q=colorbar;
43 title(cb_q,'q_{i}')
44 hold on
45 plot(X,Y,'k');
46 plot(Y,X,'k');
47 title('Uniform')
48 xlabel('Long.')
```

```
ylabel('Lat.')
print('Discrete_uni_uni','-dpng')
```

C.1.5 Discrete Search Function

```
function [ pd,n,imax,Pd_cum2 ,k] = discrete_search2(p,q,nT,
    stop )
%Conducts a Discrete Search base on inputs.
%p=Location probability density
%q=Conditional miss probabilit density
%nT=max number of glimpses
%stop=Probability of Detection to stop algorithm
%Outputs
%pd=updated location probability density
%n=search matrix
%imax=Last box searched
%Pd_cum=total probabiliy of detection
%k=number of glimpses
if ~exist('stop')
    stop=sum(p(:));
elseif stop>sum(p(:))
    stop=sum(p(:));
end
p0=1-sum(p(:));
n=zeros(size(p));%initialize look counter
Pd_cum=0;
Pd_cum2=Pd_cum;
pc=p;
pc_0=p0;
k=1;
while k<=nT && Pd_cum2(end)<stop
    pd=pc.*(1-q).*q.^n;
    [pmax,imax(k)]=max(pd(:));
    pmax;
```

```
        [irow,icol]=ind2sub(size(pd),imax(k));
        n(irow,icol)=n(irow,icol)+1;
        Pd_cum2(k+1)=Pd_cum2(k)+pmax;
        k=k+1;
end
pd=pc.*(1-q).*q.^n;
end
```

C.2 Bayesian

C.2.1 Main Code

```
clc
clear all
bayes_search_prob
bayes_loc_probs2
close all
%p-uniform
bayes_pUni_qUni
p0_uni_uni=pc_0
bayes_pUni_qlongstrat
p0_uni_ls=pc_0
bayes_pUni_qbothstrat
p0_uni_bs=pc_0
%p-long stratified
bayes_plongstrat_qUni
p0_ls_uni=pc_0
bayes_plongstrat_qlongstrat
p0_ls_ls=pc_0
bayes_plongstrat_qbothstrat
p0_ls_bs=pc_0
%p-both stratified
bayes_pbothstrat_qUni
```

```
p0_bs_uni=pc_0
bayes_pbothstrat_qlongstrat
p0_bs_ls=pc_0
bayes_pbothstrat_qbothstrat
p0_bs_bs=pc_0
%p-normal center
bayes_pnormwide_qUni
p0_nw_uni=pc_0
bayes_normwide_qlongstrat
p0_nw_ls=pc_0
bayes_normwide_qbothstrat
p0_nw_bs=pc_0
%p-normal side
bayes_pnormside_qUni
p0_ns_uni=pc_0
bayes_normside_qlongstrat
p0_ns_ls=pc_0
bayes_normside_qbothstrat
p0_ns_bs=pc_0
%p-normal corner
bayes_pnormc_qUni
p0_nc_uni=pc_0
bayes_normc_qlongstrat
p0_nc_uni=pc_0
bayes_normc_qbothstrat
p0_nc_bs=pc_0

pd=[Pd_uni_uni(end),Pd_uni_ls(end),Pd_uni_bs(end),Pd_ls_uni(
    end),...
      Pd_ls_ls(end),Pd_ls_bs(end),Pd_bs_uni(end),Pd_bs_ls(end)
          ,Pd_bs_bs(end)...
      ,Pd_nw_uni(end),Pd_nw_ls(end),Pd_nw_bs(end),Pd_ns_uni(
          end),...
```

```
52      Pd_ns_ls(end),Pd_ns_bs(end),Pd_nc_uni(end),Pd_nc_ls(end)
            ,...
53      Pd_nc_bs(end)]
54  %%
55
56  close all
57  set(0,'defaulttextinterpreter','latex')
58  [ ~,~,~,Pd_uni_uni,k,~,~] = bayes_search(P_uni,q_uni,500,
        .95 );
59  [ ~,~,~,Pd_uni_ls,k,~,~] = bayes_search(P_uni,q_long_strat
        ,500, .95 );
60  [ ~,~,~,Pd_uni_bs,k,~,~] = bayes_search(P_uni,q_both_strat
        ,500, .95 );
61  figure
62  step1=10:round(.05*length(Pd_uni_uni)):length(Pd_uni_uni);
63  step2=10:round(.05*length(Pd_uni_ls)):length(Pd_uni_ls);
64  step3=10:round(.05*length(Pd_uni_bs)):length(Pd_uni_bs);
65  plot(step1,Pd_uni_uni(step1),'xk',step2,Pd_uni_ls(step2),'xr
        ',step3,Pd_uni_bs(step3),'xg')
66  g = fittype('a-exp(b*x)','dependent','Prob','independent','x
        ');
67
68  [fit_bb1,gof_bb1]=fit(step1',Pd_uni_uni(step1)',g,'
        StartPoint',[0,0],'Upper',[1,0]);
69  [fit_bb2,gof_bb2]=fit(step2',Pd_uni_ls(step2)',g,'StartPoint
        ',[0,0],'Upper',[1,0]);
70  [fit_bb3,gof_bb3]=fit(step3',Pd_uni_bs(step3)',g,'StartPoint
        ',[0,0],'Upper',[1,0]);
71  hold on
72  plot(fit_bb1,'k')
73  plot(fit_bb2,'r')
74  plot(fit_bb3,'g')
75  xlabel('n')
```

```
76 ylabel('$P_{D}$')
77 ylim([0,1])
78 title('$P_{D}$ vs Number of Glimpses and Exponential Fitted
     Curves')
79 legend('(p_{uniform},q_{uniform})','(p_{uniform},q_{strat a
     })','(p_{uniform},q_{strat b})'...
80       ,sprintf('%.5f-e^{%.5f*x}',fit_bb1.a,fit_bb1.b),sprintf(
           '%.5f-e^{%.5f*x}',fit_bb2.a,fit_bb2.b),...
81       sprintf('%.5f-e^{%.5f*x}',fit_bb3.a,fit_bb3.b),'location
           ','best')
82
83 print('Bayes_Uni_Fit','-dpng')
84 %%
85
86 [~,~,~,Pd_ls_uni,k,~,~] = bayes_search(P_long_strat,q_uni
     ,500, .95 );
87 [~,~,~,Pd_ls_ls,k,~,~] = bayes_search(P_long_strat,
     q_long_strat,500, .95 );
88 [~,~,~,Pd_ls_bs,k,~,~] = bayes_search(P_long_strat,
     q_both_strat,500, .95 );
89 figure
90 step1=10:round(.05*length(Pd_ls_uni)):length(Pd_ls_uni);
91 step2=10:round(.05*length(Pd_ls_ls)):length(Pd_ls_ls);
92 step3=10:round(.05*length(Pd_ls_bs)):length(Pd_ls_bs);
93 plot(step1,Pd_ls_uni(step1),'xk',step2,Pd_ls_ls(step2),'xr',
     step3,Pd_ls_bs(step3),'xg')
94 [fit_bb4,gof_bb4]=fit(step1',Pd_ls_uni(step1)',g,'StartPoint
     ',[1,0],'Upper',[1,0]);
95 [fit_bb5,gof_bb5]=fit(step2',Pd_ls_ls(step2)',g,'StartPoint'
     ,[1,0],'Upper',[1,0]);
96 [fit_bb6,gof_bb6]=fit(step3',Pd_ls_bs(step3)',g,'StartPoint'
     ,[1,0],'Upper',[1,0]);
97 hold on
```

```
plot(fit_bb4,'k')
plot(fit_bb5,'r')
plot(fit_bb6,'g')
xlabel('n')
ylabel('$P_{D}$')
title('$P_{D}$ vs Number of Glimpses and Exponential Fitted Curves')
legend('(p_{strat a},q_{uniform})','(p_{strat a},q_{strat a})','(p_{strat a},q_{strat b})'...
    ,sprintf('%.5f-e^{%.5f*x}',fit_bb4.a,fit_bb4.b),sprintf('%.5f-e^{%.5f*x}',fit_bb5.a,fit_bb5.b),...
    sprintf('%.5f-e^{%.5f*x}',fit_bb6.a,fit_bb6.b),'location','best')
print('Bayes_Long_Fit','-dpng')

[~,~,~,Pd_bs_uni,k,~,~] = bayes_search(P_both_strat,q_uni,500, .95 );
[~,~,~,Pd_bs_ls,k,~,~] = bayes_search(P_both_strat,q_long_strat,500, .95 );
[~,~,~,Pd_bs_bs,k,~,~] = bayes_search(P_both_strat,q_both_strat,500, .95 );
figure
step1=10:round(.05*length(Pd_bs_uni)):length(Pd_bs_uni);
step2=10:round(.05*length(Pd_bs_ls)):length(Pd_bs_ls);
step3=10:round(.05*length(Pd_bs_bs)):length(Pd_bs_bs);
plot(step1,Pd_bs_uni(step1),'xk',step2,Pd_bs_ls(step2),'xr',step3,Pd_bs_bs(step3),'xg')
[fit_bb7,gof_bb7]=fit(step1',Pd_bs_uni(step1)',g,'StartPoint',[1,0],'Upper',[1,0]);
[fit_bb8,gof_bb8]=fit(step2',Pd_bs_ls(step2)',g,'StartPoint',[1,0],'Upper',[1,0]);
[fit_bb9,gof_bb9]=fit(step3',Pd_bs_bs(step3)',g,'StartPoint',[1,0],'Upper',[1,0]);
```

```
120 hold on
121 plot(fit_bb7,'k')
122 plot(fit_bb8,'r')
123 plot(fit_bb9,'g')
124 xlabel('n')
125 ylabel('$P_{D}$')
126 title('$P_{D}$ vs Number of Glimpses and Exponential Fitted
      Curves')
127 legend('(p_{strat b},q_{uniform})','(p_{strat b},q_{strat a
    })','(p_{strat b},q_{strat b})'...
128       ,sprintf('%.5f-e^{%.5f*x}',fit_bb7.a,fit_bb7.b),sprintf(
          '%.5f-e^{%.5f*x}',fit_bb8.a,fit_bb8.b),...
129       sprintf('%.5f-e^{%.5f*x}',fit_bb9.a,fit_bb9.b),'location
          ','best')
130 print('Bayes_Both_Fit','-dpng')
131
132 [~,~,~,Pd_nw_uni,k,~,~] = bayes_search(P_norm_wide,q_uni
    ,500, .95 );
133 [~,~,~,Pd_nw_ls,k,~,~] = bayes_search(P_norm_wide,
    q_long_strat,500, .95 );
134 [~,~,~,Pd_nw_bs,k,~,~] = bayes_search(P_norm_wide,
    q_both_strat,500, .95 );
135 figure
136 step1=10:round(.05*length(Pd_nw_uni)):length(Pd_nw_uni);
137 step2=10:round(.05*length(Pd_nw_ls)):length(Pd_nw_ls);
138 step3=10:round(.05*length(Pd_nw_bs)):length(Pd_nw_bs);
139 plot(step1,Pd_nw_uni(step1),'xk',step2,Pd_nw_ls(step2),'xr',
    step3,Pd_nw_bs(step3),'xg')
140 [fit_bb10,gof_bb10]=fit(step1',Pd_nw_uni(step1)',g,'
    StartPoint',[1,0],'Upper',[1,0]);
141 [fit_bb11,gof_bb11]=fit(step2',Pd_nw_ls(step2)',g,'
    StartPoint',[1,0],'Upper',[1,0]);
142 [fit_bb12,gof_bb12]=fit(step3',Pd_nw_bs(step3)',g,'
```

```
    StartPoint',[1,0],'Upper',[1,0]);
hold on
plot(fit_bb10,'k')
plot(fit_bb11,'r')
plot(fit_bb12,'g')
xlabel('n')
ylabel('$P_{D}$')
title('$P_{D}$ vs Number of Glimpses and Exponential Fitted
    Curves')
legend('(p_{norm},q_{uniform})','(p_{norm},q_{strat a})','(
    p_{norm},q_{strat b})'...
      ,sprintf('%.5f-e^{%.5f*x}',fit_bb10.a,fit_bb10.b),
         sprintf('%.5f-e^{%.5f*x}',fit_bb11.a,fit_bb11.b),...
      sprintf('%.5f-e^{%.5f*x}',fit_bb12.a,fit_bb12.b),'
         location','best')
print('Bayes_Norm_Fit','-dpng')

[~,~,~,Pd_ns_uni,k,~,~] = bayes_search(P_norm_side,q_uni
    ,500, .95 );
[~,~,~,Pd_ns_ls,k,~,~] = bayes_search(P_norm_side,
    q_long_strat,500, .95 );
[~,~,~,Pd_ns_bs,k,~,~] = bayes_search(P_norm_side,
    q_both_strat,500, .95 );
figure
step1=10:round(.05*length(Pd_ns_uni)):length(Pd_ns_uni);
step2=10:round(.05*length(Pd_ns_ls)):length(Pd_ns_ls);
step3=10:round(.05*length(Pd_ns_bs)):length(Pd_ns_bs);
plot(step1,Pd_ns_uni(step1),'xk',step2,Pd_ns_ls(step2),'xr',
    step3,Pd_ns_bs(step3),'xg')
[fit_bb13,gof_bb13]=fit(step1',Pd_ns_uni(step1)',g,'
    StartPoint',[1,0],'Upper',[1,0]);
[fit_bb14,gof_bb14]=fit(step2',Pd_ns_ls(step2)',g,'
    StartPoint',[1,0],'Upper',[1,0]);
```

```
165 [fit_bb15,gof_bb15]=fit(step3',Pd_ns_bs(step3)',g,'
    StartPoint',[1,0],'Upper',[1,0]);
166 hold on
167 plot(fit_bb13,'k')
168 plot(fit_bb14,'r')
169 plot(fit_bb15,'g')
170 xlabel('n')
171 ylabel('$P_{D}$')
172 title('$P_{D}$ vs Number of Glimpses and Exponential Fitted
    Curves')
173 legend('(p_{norm b},q_{uniform})','(p_{norm b},q_{strat a})'
    ,'(p_{norm b},q_{strat b})'...
174     ,sprintf('%.5f-e^{%.5f*x}',fit_bb13.a,fit_bb13.b),
        sprintf('%.5f-e^{%.5f*x}',fit_bb14.a,fit_bb14.b),...
175     sprintf('%.5f-e^{%.5f*x}',fit_bb15.a,fit_bb15.b),'
        location','best')
176 print('Bayes_NormSide_Fit','-dpng')
177
178 [~,~,~,Pd_nc_uni,k,~,~] = bayes_search(P_norm_corner,q_uni
    ,500, .95 );
179 [~,~,~,Pd_nc_ls,k,~,~] = bayes_search(P_norm_corner,
    q_long_strat,500, .95 );
180 [~,~,~,Pd_nc_bs,k,~,~] = bayes_search(P_norm_corner,
    q_both_strat,500, .95 );
181 figure
182 step1=10:round(.05*length(Pd_nc_uni)):length(Pd_nc_uni);
183 step2=10:round(.05*length(Pd_nc_ls)):length(Pd_nc_ls);
184 step3=10:round(.05*length(Pd_nc_bs)):length(Pd_nc_bs);
185 plot(step1,Pd_nc_uni(step1),'xk',step2,Pd_nc_ls(step2),'xr',
    step3,Pd_nc_bs(step3),'xg')
186 [fit_bb16,gof_bb16]=fit(step1',Pd_nc_uni(step1)',g,'
    StartPoint',[1,0],'Upper',[1,0]);
187 [fit_bb17,gof_bb17]=fit(step2',Pd_nc_ls(step2)',g,'
```

```
    StartPoint',[1,0],'Upper',[1,0]);
[fit_bb18,gof_bb18]=fit(step3',Pd_nc_bs(step3)',g,'
    StartPoint',[1,0],'Upper',[1,0]);
hold on
plot(fit_bb16,'k')
plot(fit_bb17,'r')
plot(fit_bb18,'g')
xlabel('n')
ylabel('$P_{D}$')
title('$P_{D}$ vs Number of Glimpses and Exponential Fitted
    Curves')
legend('(p_{norm c},q_{uniform})','(p_{norm c},q_{strat a})'
    ,'(p_{norm c},q_{strat b})'...
    ,sprintf('%.5f-e^{%.5f*x}',fit_bb16.a,fit_bb16.b),
        sprintf('%.5f-e^{%.5f*x}',fit_bb17.a,fit_bb17.b),...
    sprintf('%.5f-e^{%.5f*x}',fit_bb18.a,fit_bb18.b),'
        location','best')
print('Bayes_NormCorner_Fit','-dpng')
format shortG
b=[fit_bb1.b,fit_bb2.b,fit_bb3.b,fit_bb4.b,fit_bb5.b,fit_bb6
    .b,fit_bb7.b,...
    fit_bb8.b,fit_bb9.b,fit_bb10.b,fit_bb11.b,fit_bb12.b,
        fit_bb13.b,fit_bb14.b,...
    fit_bb15.b,fit_bb16.b,fit_bb17.b,fit_bb18.b]
r2=[gof_bb1.rsquare,gof_bb2.rsquare,gof_bb3.rsquare,gof_bb4.
    rsquare,...
    gof_bb5.rsquare,gof_bb6.rsquare,gof_bb7.rsquare,gof_bb8.
        rsquare,...
    gof_bb9.rsquare,gof_bb10.rsquare,gof_bb11.rsquare,
        gof_bb12.rsquare,...
    gof_bb13.rsquare,gof_bb14.rsquare,gof_bb15.rsquare,
        gof_bb16.rsquare,...
    gof_bb17.rsquare,gof_bb18.rsquare]
```

```
209
210 %%
211 [ ~,~,~,Pd_uni_bs,k,~,~] = bayes_search(P_uni,q_both_strat
    ,500, 1);
212 [ ~,~,~,Pd_uni_bs_r ,k] = bayes_search_random(P_uni,
    q_both_strat,500, 1 );
213 figure
214 plot(1:length(Pd_uni_bs),Pd_uni_bs,'xr',1:length(Pd_uni_bs_r
    ),Pd_uni_bs_r,'r')
215 %%
216 for i=1:9
217     p0=.1*i;
218     P_uni=ones(10,10);
219     P_uni=(1-p0)*P_uni/sum(P_uni(:));
220     [ ~,~,~,Pd_uni_bs_p0(i,:),k,~,~] = bayes_search(P_uni,
        q_both_strat,500, 1);
221 end
222 figure
223 for i=1:9
224     hold on
225     plot(1:25:500,Pd_uni_bs_p0(i,1:25:500))
226 end
```

C.2.2 Sample code for the Uniform/Uniform case

```
1 [ pc,n,imax,Pd_uni_uni ,k,pc_0,k_check] = bayes_search(P_uni
    ,q_uni,50, .95 );
2 figure
3 subplot(2,2,1)
4 imagesc([-4.5,4.5],[-4.5,4.5],pc,[0,max(pc(:))])
5 set(gca, 'YDir', 'normal')
6 cb_p=colorbar;
7 title(cb_p,'p_{i}')
8 hold on
```

```
plot(X,Y,'k');
plot(Y,X,'k');
title('Remainder Prob.')
xlabel('Long.')
ylabel('Lat.')

subplot(2,2,2)
imagesc([-4.5,4.5],[-4.5,4.5],n,[0,max(n(:))])
set(gca, 'YDir', 'normal')
cb_p=colorbar;
title(cb_p,'N')
hold on
plot(X,Y,'k');
plot(Y,X,'k');
title('Actual Search')
xlabel('Long.')
ylabel('Lat.')
subplot(2,2,3)
imagesc([-4.5,4.5],[-4.5,4.5],P_uni,[0,max(P_uni(:))])
set(gca, 'YDir', 'normal')
cb_p=colorbar;
title(cb_p,'p_{i}')
hold on
plot(X,Y,'k');
plot(Y,X,'k');
title('Uniform')
xlabel('Long.')
ylabel('Lat.')


subplot(2,2,4)
imagesc([-4.5,4.5],[-4.5,4.5],q_uni,[0,max(q_uni(:))])
set(gca, 'YDir', 'normal')
```

```
cb_q=colorbar;
title(cb_q,'q_{i}')
hold on
plot(X,Y,'k');
plot(Y,X,'k');
title('Uniform')
xlabel('Long.')
ylabel('Lat.')
print('Bayes_uni_uni','-dpng')
```

C.2.3 Bayes Search Function

```
function [ pd,n,imax,Pd_cum ,k,pc_0,k_check] = bayes_search(
    p,q,nT, stop )
%Conducts a Bayesian Search base on inputs.
%p=Location probability density
%q=Conditional miss probabilit density
%nT=max number of glimpses
%stop=Probability of Detection to stop algorithm
%Outputs
%pd=updated location probability density
%n=search matrix
%imax=Last box searched
%k=number of glimpses
%pc_0=final probability tgt is not in search area
%k_check=game theory detection glimpse
if ~exist('stop')
    stop=sum(p(:));
elseif stop>sum(p(:))
    stop=sum(p(:));
end
check=0;
n=zeros(size(p));%initialize look counter
Pd_cum=0;
```

```
pc_0=1-sum(p(:));
pc=p;
k=1;
tgt_pos=randsample(1:101,1,true,[p(:);pc_0]);
[tgt_row,tgt_col]=ind2sub(size(p),tgt_pos);
while k<=nT && Pd_cum(end)<stop
    pd=pc.*(1-q);
    [pmax,imax(k)]=max(pd(:));
    [irow,icol]=ind2sub(size(pd),imax(k));
    if check==0
        if irow==tgt_row && icol==tgt_col
            check=binornd(1,1-q(irow,icol));
            if check==1
                k_check=k;
            end
        end
    end
    pc(irow,icol)=pc(irow,icol).*q(irow,icol);
    pc=pc/(1-pmax);
    pc_0=pc_0/(1-pmax);
    n(irow,icol)=n(irow,icol)+1;
    Pd_cum(k+1)=Pd_cum(k)+(1-Pd_cum(k))*pmax;
    k=k+1;
end
if check==0
    k_check=NaN;
end
pd=pc;
end
```

C.3 Continuous

Code is as in [6] with the exception that $R_1 = R_0 e^{-v_1/V}$ and $R_2 = R_0 e^{-v_2/V}$

THIS PAGE INTENTIONALLY LEFT BLANK

List of References

[1] CNA - About Us. (2017). Center for Naval Analyses. [Online]. Available: https://www.cna.org/about/

[2] K. Tidman, *The Operations Evaluation Group: A History of Naval Operations Analysis*, 1st ed. Annapolis, MD: Naval Institute Press, 1984.

[3] L. D. Stone, "Theory of optimal search," in *Mathematics in Science and Engineering*, vol. 118, 1st ed. New York, NY: Academic Press, Inc, 1975.

[4] H. Richardson and J. H. Discneza, "The united states coast guard computer-assisted search planning system (casp)." *Naval Research Logistics Quarterly*, vol. 27, no. 4, December.

[5] B. O. Koopman, "Search and screening," Operations Evaluation Group, Washington, DC, Tech. Rep. OEG-56.

[6] Z. C. Lukens, "Optimizing search patterns for multiple searchers prosecuting a single contact in the south china sea," Master's thesis, Dept. of Applied Mathematics, Naval Postgraduate School, 1 University Cir, Monerey, CA, 2016.

[7] V. Sozen, "Optimal deployment of unmanned aerial vehicles for border surveillance," Master's thesis, Dept. Ops. Research, Naval Postgraduate School, 1 University Cir, Monerey, CA, 2014.

[8] M. Atkinson, M. Kress, and R.-J. Lange, "When is information sufficient for action? search with unreliable yet informative intelligence," *Operations Research*, vol. 64, no. 2, p. 315–328, 2016.

[9] H. Wang and H. Zhou, "Computational studies on detecting a diffusing target in a square region by a stationary or moving searcher," *American Journal of Operations Research*, vol. 5, no. 2, pp. 47–68, March 2015.

[10] H. Wang and H. Zhou, "Searching for a target traveling between a hiding area and an operating area over multiple routes," *American Journal of Operations Research*, vol. 5, no. 4, pp. 258–273, July 2015.

[11] H. Wang and H. Zhou, "Performance of stochastically intermittent sensors in detecting a target traveling between two areas," *American Journal of Operations Research*, vol. 6, no. 2, pp. 199–212, March 2016.

[12] A. R. Washburn, *Search and Detection*, 5th ed. Catonsville, MD: Institute for Operations Research and the Management Sciences, 2014.

[13] J. Eagle, "Search detection theory," (OA3602) Lecture Notes, Dept. of Operations Research, Naval Postgraduate School, Monterey, CA, spring 2015.

[14] A. R. Washburn and M. Kress, *Combat Modeling (International Series in Operations Research and Management Science)*. New York: Springer US, 2009, vol. 134.

[15] C. M. Keller, "Bayesian search for missing aircraft," Fusion 2015 Plenary Talk, Metron Inc., www.metsci.com, 2015.

[16] D. B. K. J. Arrow and M. A. Girshick, "Bayes and minimax solutions of sequential decision problems," *Econometrica*, vol. 17, no. 3-4, pp. 213–244, 1949.

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California